

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Desarrollo de métodos de evaluación de algoritmos de descomposición de EMG



Grado en Ingeniería
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Estibaliz Olaechea Olaechea

Javier Navallas Irujo

Pamplona, 28 de Junio de 2017

Abstract

The purpose of this project is the design and implementation of methods to evaluate the quality of an automatic EMG decomposition system. These methods will allow to evaluate jointly and separately the performance of the processes of segmentation and classification of different algorithms that we have available. Different experiments will be carried out for different muscle activity levels in order to observe the influence of some parameters of these algorithms on the quality of the decomposition system. The results of the evaluation will allow us to implement the necessary changes in the algorithm in order to improve its quality.

Keywords

EMG decomposition; Segmentation; Classification; Firing Patterns;

Resumen

La finalidad de este proyecto es el diseño e implementación de métodos que nos permitan evaluar la calidad de un sistema de descomposición automática EMG. Estos métodos van a permitir evaluar de forma conjunta y por separado el rendimiento de los procesos de segmentación y clasificación de diferentes algoritmos que tenemos disponibles. Se realizarán para distintos niveles de contracción diversos experimentos con el fin de observar la influencia de algunos parámetros de estos algoritmos en la calidad del sistema de descomposición. Los resultados de la evaluación nos permitirán implementar los cambios necesarios en los distintos algoritmos para mejorar su calidad.

Palabras clave

Descomposición EMG; Segmentación; Clasificación; Trenes de disparo;

ÍNDICE

Capítulo 1. Introducción.....	6
1.1 Contexto académico.....	6
1.2 Objetivo del proyecto.....	6
1.3 Organización de la memoria	6
Capítulo 2. Introducción a la simulación de señales EMG	7
2.1 Modelo del músculo.....	7
2.1.1 Geometría del músculo esquelético.....	7
2.1.2 La unidad motora	8
2.1.3 Distribución espacial de los territorios de las MUs.	9
2.1.4 Distribución espacial de las fibras de las MUs	10
2.1.5 Zona de inervación y fracciones	11
2.1.6. Velocidad de conducción	12
2.1.7. Reclutamiento de las MUs	13
2.1.8. Trenes de disparo	13
2.1.9. Simulación modelo del músculo	14
2.2 Modelo del electrodo.....	15
2.3 Obtención del MUP	16
2.4 Obtención de trenes de disparo.....	16
2.5 Obtención de la señal EMG	16
Capítulo 3. Introducción a la descomposición de señales EMG	18
3.1 Descomposición manual	19
3.2 Descomposición automática	23
Capítulo 4. Evaluación de los sistemas de clasificación automática.....	24
4.1 Preprocesado	24
4.2 Segmentación.....	25
4.2.1 Detección.....	25
4.2.2 Búsqueda de los máximos.....	26
4.2.3 Exclusión de máximos	27
4.2.4 Enventanado de los segmentos	30
4.2.5 Instantes de disparo	30
4.3 Evaluación de la segmentación	31
4.3.1 Determinación del rendimiento de la segmentación.....	31
4.3.2 Mejoras del algoritmo de segmentación	34

4.3.3 Variación del tamaño de la ventana de exclusión.....	38
4.3.4 Variación de MVC.....	44
4.4 Clasificación.....	47
4.4.1.Extracción de características.....	47
4.4.2 Clasificación no supervisada.....	47
4.4.3 Obtención de prototipos y patrones de disparo.....	47
4.5 Evaluación de la clasificación	48
4.5.1 Descripción del experimento	52
4.5.2 Influencia de las superposiciones.....	52
4.5.3 Algoritmos de clasificación.....	53
4.5.4 Comparación de los diferentes algoritmos de clasificación.....	54
4.6 Evaluación del proceso completo: segmentación y clasificación.....	58
4.6.1 Descripción del experimento	59
4.6.2 Influencia de los errores de la segmentación	59
4.6.3 Influencia de las superposiciones.....	60
4.6.4 Comparación de los distintos algoritmos.....	60
Capítulo 5. Conclusiones y líneas futuras	65
5.1 Conclusiones.....	65
5.2 Líneas futuras	66
Capítulo 6. Bibliografía.....	67

Capítulo 1. Introducción

1.1 Contexto académico

Este trabajo fin de grado se ha desarrollado en la Universidad Pública de Navarra, concretamente en el área de Teoría de la Señal y Comunicaciones, dentro del Grupo de Investigación de Ingeniería Biomédica.

1.2 Objetivo del proyecto

El objetivo principal de este proyecto es el diseño e implementación de métodos que nos ayuden a evaluar la calidad de un sistema de descomposición automática EMG. A partir de los diferentes algoritmos que disponemos para llevar a cabo el proceso de descomposición, se realizan diversos experimentos que nos permiten desarrollar el método más apropiado para evaluar de forma conjunta y por separado el rendimiento de los procesos de segmentación y clasificación. Una vez desarrollados, los resultados de la evaluación nos permitirán implementar las mejoras que requiera el algoritmo en cuestión.

1.3 Organización de la memoria

En el capítulo 2 se realizará una introducción del *toolbox* de simulación que se utilizará a lo largo del proyecto describiendo el proceso completo que se lleva a cabo para la formación de la señal EMG. En el capítulo 3, se introducen los procesos involucrados en los sistemas de descomposición de una señal EMG. Se realizará además un marco de evaluación sobre el desempeño de sistemas semiautomáticos o manuales como EMGLab y sistemas automáticos como Montreal. En el capítulo 4, se describe el funcionamiento del sistema de descomposición automática implementado en este proyecto y los métodos desarrollados para su evaluación. Con dichos métodos, se evaluarán diferentes algoritmos para determinar cuál resulta más acertada. Finalmente, se concluye con el capítulo 5 donde se presentan las conclusiones y líneas futuras.

Capítulo 2. Introducción a la simulación de señales EMG

En este apartado se hará una introducción al funcionamiento del *toolbox* de simulación de señales electromiográficas (EMG, del inglés *electromyography*) desarrollado en Matlab por el equipo de investigación de Ingeniería Biomédica de la UPNA, el cual se utilizará a lo largo del proyecto.

Las señales EMG son señales eléctricas que miden el potencial eléctrico generado por las fibras musculares durante su contracción. El proceso de formación de la señal EMG (funciones `sim_emg`) se lleva a cabo simulando previamente el modelo del músculo (funciones `sim_muscle_`) y del electrodo para después simular los potenciales de unidad motora (funciones `sim_mup_`) y trenes de disparo (funciones `sim_firing_`). Este proceso se realiza empleando una serie de estructuras de datos y funciones que se muestran en el diagrama de bloques de la Fig. 1. En este capítulo, veremos en que consiste cada uno de ellos.

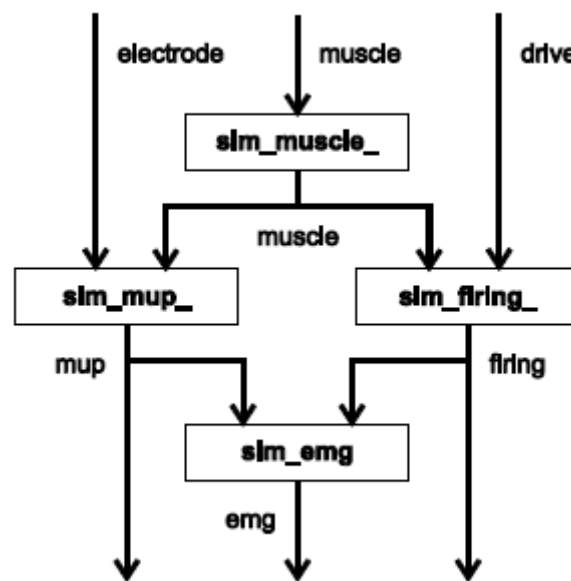


Figura 1. Diagrama de bloques de la simulación de una señal EMG

2.1 Modelo del músculo

La simulación del músculo se realiza definiendo previamente nuestro modelo a partir de una serie de parámetros en la estructura `[muscle]` que veremos a continuación:

2.1.1 Geometría del músculo esquelético

El modelo del músculo con el que trabajaremos, sigue una geometría cilíndrica cuyas dimensiones se establecen mediante un vector llamado *dim*, donde indicaremos el radio de la sección transversal del músculo y la longitud del mismo en mm. Si

queremos tener en cuenta la capa de piel y de grasa que encontramos por encima del músculo, disponemos de los parámetros *skin* y *fat* en los que introduciremos el grosor adecuado también en mm.

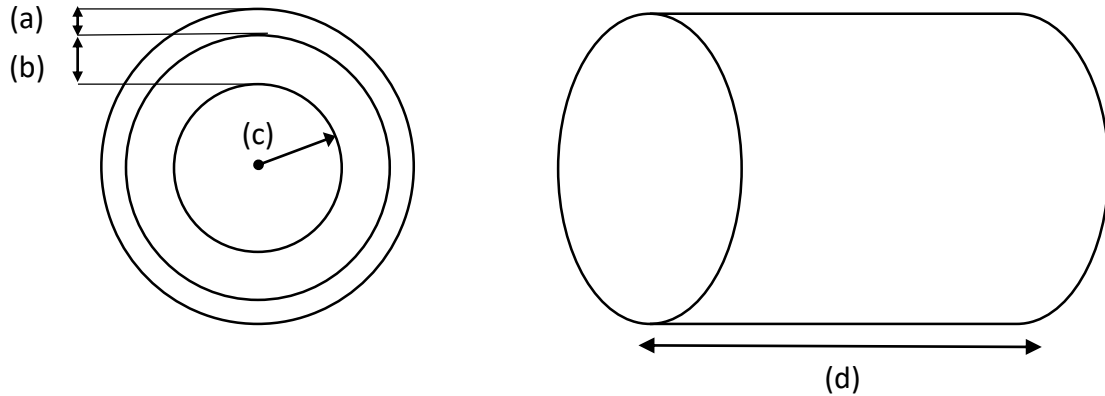


Figura 2. Geometría cilíndrica del modelo del músculo esquelético. Sección transversal y longitudinal. (a) *muscle.skin* (b) *muscle.fat* (c) *muscle.dim(1)* (d) *muscle.dim(2)*.

2.1.2 La unidad motora

Una vez fijado el tamaño del músculo, el siguiente paso que daremos será determinar los parámetros relacionados con las unidades motoras.

La unidad motora (MU, del inglés *Motor Unit*) es la unidad anatómica y funcional de un músculo esquelético que consta de la motoneurona α con su axón y las fibras musculares inervadas por ella (Fig. 3). Asimismo, la sección del músculo en la que se encuentran las fibras musculares de dicha MU, se conoce como territorio de la unidad motora (MUT, del inglés *Motor Unit Territory*).

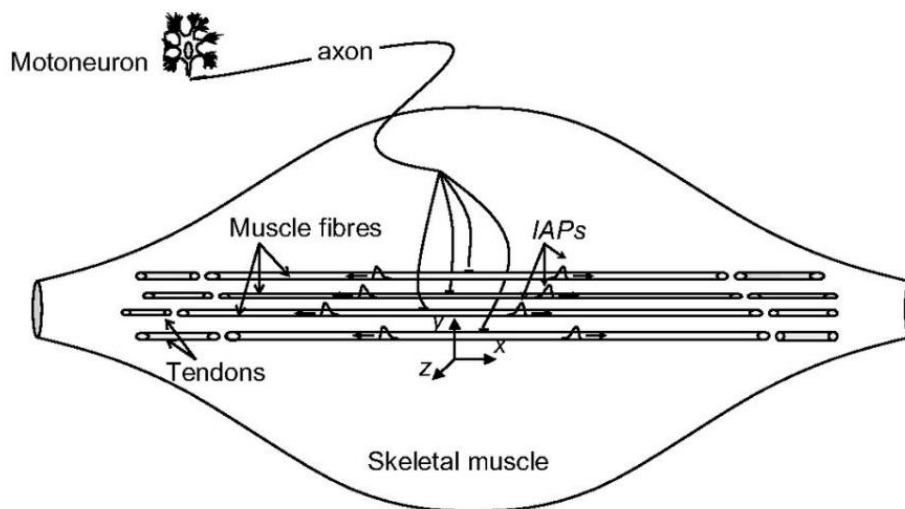


Figura 3. Representación de la unidad motora y sus componentes

La contracción muscular se genera por la activación repetida de varias MUs, cada uno de los cuales están controlados por una única motoneurona a través de su axón. En función de la fuerza ejercida por el músculo se activarán más o menos, donde las más pequeñas serán las primeras en activarse.

En nuestro modelo, empezaremos por determinar el número de unidades motoras que contiene el músculo en su totalidad y esto lo haremos mediante el parámetro *mun*. El siguiente paso, será fijar el tamaño de cada una de ellas en función del número de fibras que éstas contengan. Para ello, disponemos de dos parámetros que fijarán el número de fibras que inerva la unidad motora más pequeña y las que inerva la más grande, *mufn_min* y *mufn_max* respectivamente. A partir de estos dos valores, el número de fibras de las MUs se modela empleando una distribución exponencial implementada según el modelo de Fuglevand [2].

Otro parámetro que tendremos en cuenta para caracterizar la unidad motora es la densidad de fibras de la unidad motora (MUFD, del inglés *Motor Unit Fiber Density*), es decir, el número de fibras por unidad de superficie en mm^2 dentro del MUT. La densidad permanecerá relativamente constante para todas las MUs, debido a la baja correlación que se tiene entre el número de fibras de la unidad motora (MUFN, del inglés *Motor Unit Fiber Number*) y MUFD. Además, su valor estará determinado dentro de un rango que se define mediante dos valores que representan la media (*mufd_med*) y la varianza (*mufd_var*). De esta forma nos aseguramos que el valor de la densidad de todas las unidades motoras esté dentro del rango que hayamos definido.

A partir de MUFD y MUFN, se determina el área del territorio de la unidad motora (MUTA del inglés *Motor Unit Territory Area*) que el modelo almacena en el vector *muta*, y esto permite calcular los radios de los MUT que almacena en el vector *mutr*. El aumento del tamaño del territorio de la MU está relacionado con el aumento del MUFN, habiendo una fuerte correlación entre ambas. Esto implica que la MUTA también seguirá una distribución exponencial.

2.1.3 Distribución espacial de los territorios de las MUs.

Ya tenemos definidos los tamaños de todas las MUs, por lo que ahora trataremos de ubicarlas. Para colocarlas, se aplica un algoritmo que busca una disposición tal que satisfaga un solapamiento constante entre los territorios de las MUs. Mediante el parámetro *delta* podremos elegir la distancia entre los puntos en los que el algoritmo calcula el solapamiento y permite ubicar los MUTs (Fig. 4.).

Se permite que los territorios excedan parcialmente el límite del músculo, esas partes externas se cortan y el radio del territorio que ha sido cortado se aumenta de tal forma que el área de la región del territorio, MUTA, sea la misma que la original.

En la Fig. 4 podemos observar las regiones de los territorios originales como círculos discontinuos y una vez que se aumenta el radio queda definido el territorio en un círculo sólido.

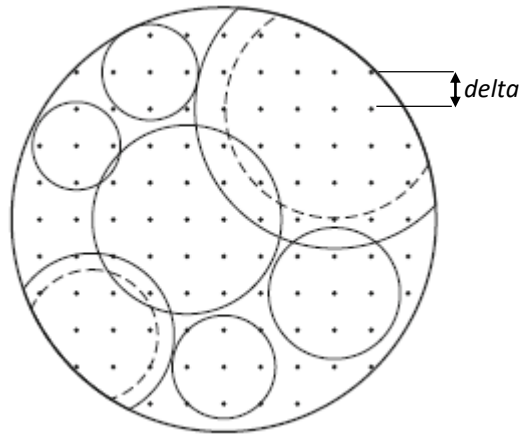


Figura 4. Representación de la colocación de los MUT

2.1.4 Distribución espacial de las fibras de las MUs

Una vez tengamos ubicados los territorios de las MUs, debemos determinar la distribución espacial de las fibras musculares. Para ello, se crea una rejilla hexagonal donde se representa el conjunto de todas las fibras musculares que se observan en la sección transversal del músculo. El área de la fibra muscular se define a través del parámetro mfa .

Después, cada fibra escoge una MU de las que solapan su posición (Fig. 5 (a)) y la identidad de esa MU se guardará en el vector $finn$. Como resultado de este proceso, cada MU tiene un número de fibras musculares inervadas dispersas dentro de su territorio (Fig. 5 (b)).

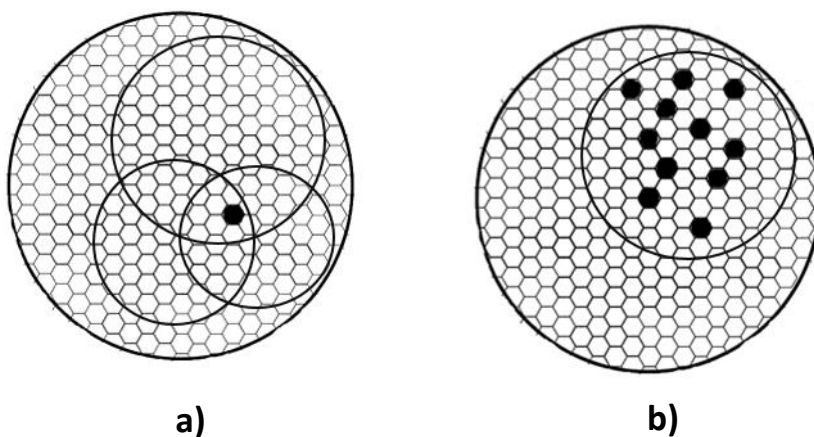


Figura 5. a) Distribución espacial de las fibras musculares, donde la fibra sombreada debe escoger una de las 3 MUs que solapan su posición. b) Una MU y las fibras musculares que inerva.

2.1.5 Zona de inervación y fracciones

Como se ha comentado anteriormente, el axón de la motoneurona al llegar al músculo se ramifica en múltiples terminales axonales cada una de las cuales alcanza una terminación nerviosa (placa motora) que se conecta a la fibra muscular. A esta unión entre el axón y la fibra se le denomina unión neuromuscular. Tras la unión neuromuscular, el potencial de acción transmitido desde la ramificación del axón hasta la fibra se propaga de forma bidireccional a lo largo de cada fibra muscular.

La zona de inervación de una unidad motora es la región que comprende las posiciones en las que se producen las uniones neuromusculares. Suele posicionarse en la zona media de las fibras musculares como se aprecia en la Fig. 6.

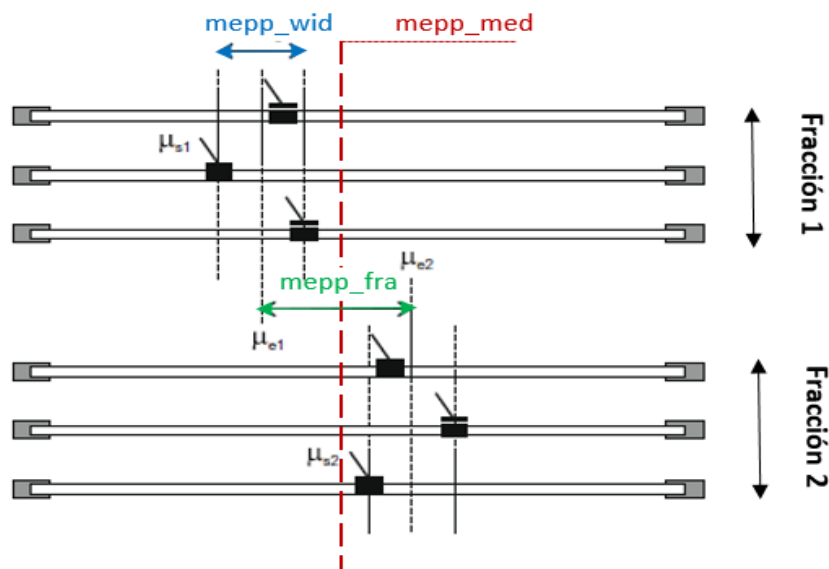


Figura 6. Representación de la distribución de las uniones neuromusculares de dos fracciones de una misma MU.

En nuestro modelo, el músculo se divide en regiones de forma hexagonal sobre las cuales se ubican las MUs, cada una con sus correspondientes fibras musculares. Las fibras correspondientes a una misma MU se encontrarán en distintas posiciones de manera que distintos grupos de estas fibras van a pertenecer a distintas regiones hexagonales. A dichas regiones les llamaremos fracciones. El tamaño de las regiones hexagonales se define mediante el parámetro fra (Fig. 7).

Para modelar la zona de inervación de la unidad motora (Fig. 6) empezaremos asignando un valor al parámetro $mepp_med$ que nos representa la media donde se producen todas las uniones neuromusculares de todas las unidades motoras. Después, con el parámetro $mepp_fra$ se determina la variabilidad de la inervación media, entre fracciones de la MU respecto de la media del músculo. Por último, fijaremos la desviación estándar dentro de cada fracción mediante $mepp_wid$.

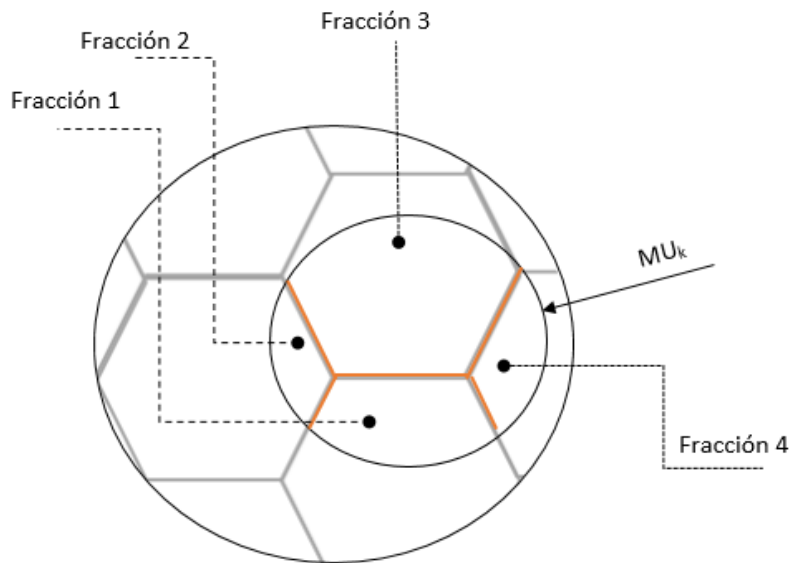


Figura 7. Representación esquemática de las fracciones. Las fibras que inerva la MU_k pertenecen a cuatro fracciones distintas.

2.1.6. Velocidad de conducción

El potencial de acción se propaga a cierta velocidad de conducción a través de la fibra muscular. Esta velocidad es directamente proporcional al diámetro de la fibra, teniendo así que las MUs pequeñas formadas por fibras pequeñas suelen tener una velocidad baja, y las MUs grandes formadas por fibras grandes tienen una velocidad mayor.

La velocidad de conducción de las distintas MUs de nuestro músculo seguirá una distribución exponencial donde se fijará la velocidad correspondiente a la MU más pequeña con el parámetro $mucv_min$ y el de la MU más grande con $mucv_max$.

Una vez tengamos definida la velocidad de conducción para cada MU, determinaremos la velocidad de las fibras musculares dándoles un valor en torno a una media, que será el valor de la velocidad de su correspondiente MU, y un coeficiente de variación fijado mediante el parámetro $mucv_cov$. De esta forma, cada fibra muscular de una misma MU tendrá una velocidad distinta pero cercana a la de su MU correspondiente.

El potencial de acción que proviene del axón de la motoneurona llega a la unión neuromuscular en distintos instantes de tiempo produciendo un retardo que podemos establecer mediante el parámetro $miod$. El retardo de propagación depende principalmente de la velocidad de conducción y de la longitud de la rama terminal del axón por donde se propaga. Normalmente, el valor de este retardo es despreciable debido a que la velocidad de propagación del impulso a través del axón suele ser muy alta y la diferencia de longitudes de las ramificaciones no son demasiado grandes.

2.1.7. Reclutamiento de las MUs

El reclutamiento es la forma en que el sistema nervioso central selecciona las unidades motoras a ser activadas cuando se requiere una determinada fuerza muscular.

El nivel de contracción medido como porcentaje respecto del nivel de contracción voluntario máximo (MVC, del inglés *Maximum Voluntary Contraction*), está directamente relacionado con la tasa de disparo y el número de unidades motoras reclutadas. Cuanto mayor es el porcentaje de contracción respecto del MVC, mayor será la tasa de disparo y también el número de MUs reclutadas, siendo los primeros reclutados las unidades motoras más pequeñas y los últimos las más grandes. Las primeras MUs reclutadas generalmente tendrán mayor frecuencia de disparo que las reclutadas posteriormente.

A partir del parámetro fr_{rrn} se calcula el rango de reclutamiento, es decir el umbral de excitación de reclutamiento (RTE, del inglés *Recruitment Threshold Excitation*) de la unidad motora de mayor índice. Este valor nos indica también el porcentaje del nivel de MVC necesaria para llegar a reclutar la última MU. Cuanto menor sea su valor, antes se reclutan todas las MUs y por tanto, antes se alcanzarán altos valores de fuerza muscular.

El parámetro fr_{rte} , hace referencia al nivel mínimo de excitación necesario para que la MU comience a desarrollar su actividad. A partir del índice de cada MU, podremos observar su valor de RTE correspondiente. Este valor, sigue una distribución exponencial, de manera que cuanto mayor es el índice de la MU mayor es su valor de RTE.

2.1.8. Trenes de disparo

La tasa de disparo (FR, del inglés *Firing Rate*), es el número de impulsos por segundo que llegan desde la motoneurona hasta las uniones neuromusculares de las fibras que inerva. Para mantener la contracción muscular en el tiempo es necesario que las MUs disparen repetidamente generando un tren de disparos.

La diferencia temporal entre dos disparos se ha modelado a partir de una distribución gaussiana con una media que se obtiene mediante las curvas de disparo del modelo de disparo concreto (Fuglevand o de Luca) y un coeficiente de variación que se puede fijar mediante el parámetro fr_{cov} . De esta forma nos aseguramos que a partir de un disparo, el siguiente se produzca antes de un determinado tiempo.

En la Fig. 8 se muestra el modelo de reclutamiento que propone de Luca [3]. Este modelo aproxima el comportamiento de las MUs mediante curvas, donde cada curva corresponde a una MU. Para conocer el número de MUs reclutadas y a qué frecuencia de trabajo se encuentra cada una, trazaremos una línea vertical al nivel de intensidad deseado. Las líneas que se crucen con la vertical serán aquellas MUs reclutadas siendo el punto de corte su tasa de disparo de trabajo.

Como podemos ver, la tasa de disparo de trabajo para las MUs recién reclutadas es bajo, pero una vez ya reclutados conforme aumenta la intensidad la frecuencia aumenta.

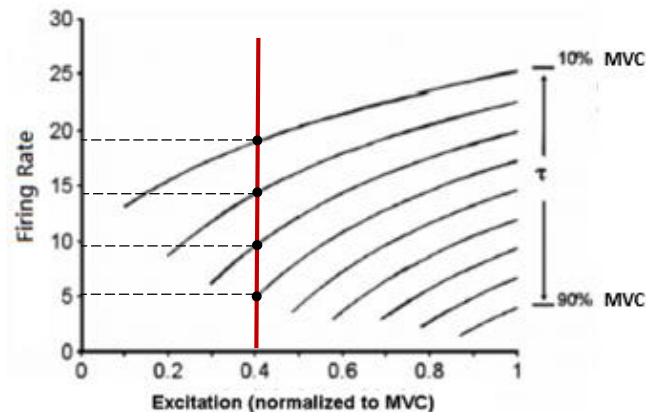


Figura 8. Modelo de reclutamiento propuesto por de Luca. Representación de la tasa de disparo de unidades motoras reclutadas entre el 10 y el 90% a intervalos de 10% de MVC.

2.1.9. Simulación modelo del músculo

Una vez hayamos definido todos los parámetros en la estructura `[muscle]`, ya tenemos formado el modelo del músculo, por lo que ahora se realizará su simulación mediante la función `sim_muscle_dsvm_inner_factions`.

Tras la simulación del músculo, obtenemos los campos de entrada y salida en la estructura de datos `[muscle]` que se detallan a continuación:

Campos de entrada

- *dim* Vector de las dimensiones del músculo; radio y longitud (mm)
- *skin* Espesor de la capa de piel (mm)
- *fat* Espesor de la capa de grasa (mm)
- *mun* Nº de unidades motoras
- *mufn_min* Nº de fibras de la MU más pequeña
- *mufn_max* Nº de fibras de la MU más grande
- *mufd_med* Vector de la densidad de fibras media de las MUs
- *mufd_var* Vector de la varianza de la densidad de fibras de las MUs
- *mfa* Área de las fibras musculares
- *frc* Tamaño de las fracciones
- *delta* Distancia puntos
- *mepp_med* Media de la zona de inervación
- *mepp_fra* Dispersión de la zona de inervación intra-fracción
- *mepp_wid* Dispersión de la zona de inervación
- *mten_wid* Dispersión de la unión músculo-tendiosa
- *mucv_min* Velocidad de conducción de la MU más pequeña

- *mucv_max* Velocidad de conducción de la MU más grande
- *mucv_cov* Coeficiente de variación de la velocidad de conducción de las MUs
- *miod* Dispersión del inicio de la depolarización
- *fr_rrn* Rango de reclutamiento
- *fr_cov* Coeficiente de variación
- *fr_mod* Modelo de “Fuglevand” o “de Luca”
- *fr_mus* Músculo concreto (parámetros predefinidos)
- *fr_rte:* Umbral de reclutamiento de la excitación

Campos de salida

- *mufn* Vector del nº de fibras de las MUs
- *mf n* Nº de fibras musculares
- *mfc* *Tamaño de las fracciones de la MU*
- *fpos* Matriz de posiciones de las fibras musculares
- *mufc* *Matriz de posiciones de las fracciones de las MUs*
- *mufd_med_* Vector densidad de fibras media de las MUs
- *mufd_var* Vector varianza de la densidad de fibras de las MUs
- *mufd_min* Densidad de fibras de la MU más pequeña
- *mufd* Densidad de fibras de las MUs
- *mucv* Vector de velocidades de conducción de las MUs
- *mutr* Matriz de radios de los territorios de las MUs
- *muta* Matriz de áreas de los territorios de las MUs
- *mutc* Matriz de centros de los territorios de las MUs
- *finn* Vector de identidad de las MUs que inervan a las fibras
- *fden* Densidad de fibras acumulada en cada posición
- *fmep* Vector de posiciones de las uniones neuromusculares de las fibras
- *ften* Vector de posiciones de las uniones musculo-tendinosas de fibras
- *fmcv* Vector de velocidades de conducción de las fibras
- *fiod* Vector de inicios de la depolarización de las fibras

2.2 Modelo del electrodo

Para obtener los potenciales de cada unidad motora, antes debemos establecer los parámetros correspondientes del modelo del electrodo que se vaya a utilizar. En este caso, los parámetros se fijarán en la variable [*electrode*].

A continuación se detallan los campos de entrada y salida de la estructura de datos [*electrode*]:

Campos de entrada

- *type* Tipo de electrodo (concéntrico o de fibra simple)
- *fsam* Frecuencia de muestreo (Hz)

- *channels* Nº de canales o puntos de registro
- *ied* Desplazamiento del electrodo entre posiciones

Campos de salida

- *pos* Vector de posiciones de registro del electrodo

2.3 Obtención del MUP

A partir de las estructuras de datos del músculo y del electrodo se obtienen los potenciales generados por cada MU mediante la siguiente función:

```
[mup]=sim_mup_needle(muscle,electrode)
```

- *muscle*: estructura de datos que contiene la información del modelo del músculo
- *electrode*: estructura de datos que contiene la información del electrodo

2.4 Obtención de trenes de disparo

Para llegar a la simulación de los trenes de disparo de las MUs, como se muestra en el esquema de la Fig. 1, junto al modelo del músculo simulado necesitamos definir el nivel de contracción máxima voluntaria con el parámetro *MVC*. Para simular los trenes de disparo se utiliza la siguiente función:

```
firing=sim_firing_ipfm_constant(muscle,drive,t_max)
```

- *muscle*: estructura de datos que contiene la información del modelo del músculo
- *drive*: porcentaje del nivel de máxima contracción voluntaria (*MVC*)
- *t_max*: margen de tiempo en el que puede variar el instante de disparo (en ms).

2.5 Obtención de la señal EMG

Por último, una vez que tenemos simulados el modelo del músculo y del electrodo junto a los trenes de disparo, se obtiene la señal EMG. Para su obtención se realiza el sumatorio de la convolución entre los trenes de disparo de las MUs detectadas por el electrodo con sus correspondientes potenciales (MUP, del inglés *Motor Unit Potential*) obtenidos. Este proceso se realiza mediante la siguiente función:

```
emg=sim_emg_noint(mup,firing,electrode,t_max)
```

- *mup*: estructura *cell-array* que contiene los MUPs ideales de cada MU, sin interferencias de otras unidades motoras.
- *firing*: estructura *cell-array* que contiene los instantes de disparo ideales (en ms) de las unidades motoras del músculo.
- *electrode*: estructura de datos que contiene la información del electrodo
- *t_max*: margen de tiempo en el que puede variar el instante de disparo (en ms).

Finalmente, a la señal obtenida, se le suma cierto ruido de fondo generado por interferencias de otros músculos, simulado como un ruido aditivo blanco gaussiano de media nula con cierta potencia definida por el usuario.

Capítulo 3. Introducción a la descomposición de señales EMG

En este capítulo, se hará una introducción a los procesos involucrados en la descomposición de una señal EMG y cómo se llevan a cabo mediante sistemas de descomposición semiautomática o manual (EMGLab) y automática (Montreal). El capítulo termina con un marco de evaluación sobre el desempeño de los algoritmos de descomposición con señales simuladas.

La descomposición de una señal EMG consiste en detectar e identificar cada potencial de todas las MUs dentro del área de captación del electrodo de registro. A la salida de la descomposición se obtendrán así, tanto los MUPs correspondientes a cada MU como los patrones de disparo.

A continuación se detallan los pasos que se llevan a cabo, representados en la Fig. 13:

a) Preprocesado:

En una señal EMG, todo lo que no son los MUPs se considera como ruido. Nos encontramos con ruido de fondo y derivas de baja frecuencia en la línea de base y ruido instrumental de alta frecuencia como el causado por el movimiento del electrodo de aguja. El primer paso, será filtrar la señal EMG para facilitar la detección de los disparos mediante umbrales.

b) Segmentación:

Después del filtrado, la señal se divide en intervalos que contengan posibles MUPs de las unidades motoras. A estos intervalos se les llaman segmentos. El objetivo aquí es localizar y estimar el comienzo y el final en el intervalo de tiempo de todos estos segmentos. Esta etapa de la descomposición tiene una gran influencia en las siguientes.

Hay que tener en cuenta que un segmento puede contener un MUP o MUPs superpuestos que se intentarán identificar para poder separarlos.

c) Clasificación:

Una vez que tengamos localizados los segmentos que contienen los MUPs, el siguiente paso será identificar los segmentos que tienen una forma similar y por tanto, pertenecientes a la misma MU. Es decir, se extraerán las características de cada segmento, empleando distintas técnicas de reconocimiento de patrones, para poder clasificarlas en grupos. Esta clasificación se conoce como clasificación no supervisada.

d) Obtención de prototipos y patrones de disparo

Después de clasificar los segmentos, se obtiene un prototipo que será el que represente al conjunto de MUPs etiquetados cada uno por su correspondiente MU. Además tendremos localizados los instantes de disparo de los MUPs y por lo tanto el resultado de los patrones de disparo.

e) Resolución de superposiciones

Como ya se ha comentado, es muy posible que un segmento contenga superpuestos varias MUPs debido a que distintas motoneuronas hayan disparado en instantes de tiempo muy cercanos. Se deberá intentar separar estos potenciales superpuestos buscando combinaciones entre los distintos MUPs.

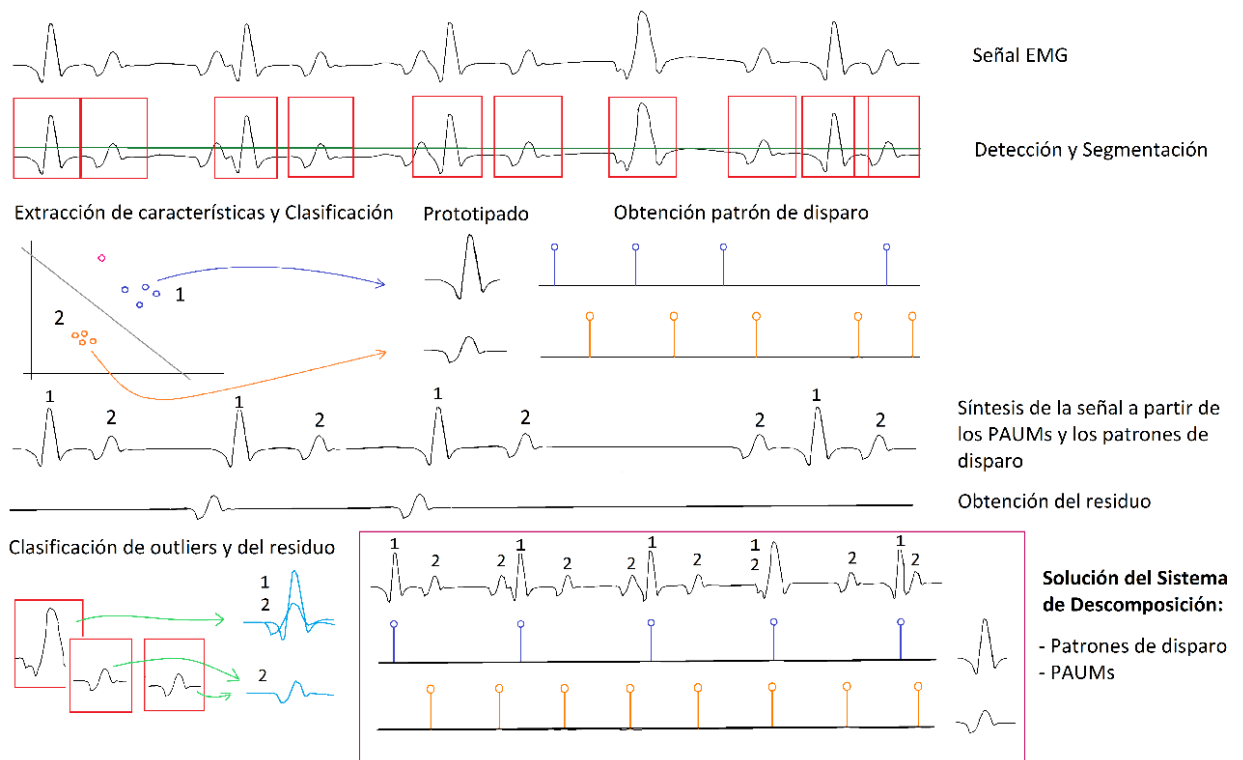


Figura 13. Proceso de descomposición de una señal EMG. [7]

3.1 Descomposición manual

En este apartado se describirá el funcionamiento del programa EMGLab, un sistema de descomposición semiautomática que permite la descomposición automática, la supervisión de los resultados y la descomposición manual. Veremos cómo se realizan todos los procesos comentados anteriormente para llevar a cabo la descomposición completa de una señal EMG.

Una vez abierto el programa, lo primero que haremos es importar la señal EMG que queremos descomponer. Después, escogeremos de la lista del menú una frecuencia de corte adecuada para el filtrado de la señal y ejecutaremos la descomposición. En este caso, se trata de una descomposición supervisada; es decir, una descomposición automática con posibilidad de edición manual. Un resultado de ejemplo es el que se muestra en la Fig. 14.

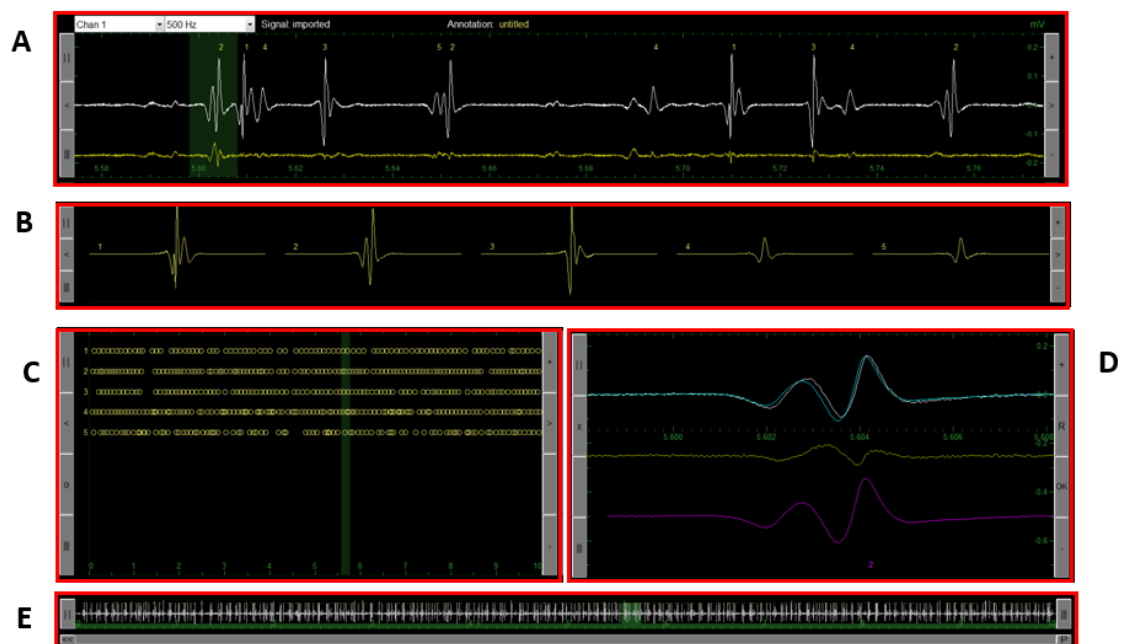


Figura 14. Interfaz gráfica del programa EMGlab. a) Representación de una parte de la señal EMG. b) Representación de los MUP detectados c) Representación del patrón de disparos de cada MU d) Ajuste de precisión e) Representación de la señal completa de EMG.

El programa nos muestra la señal EMG con los potenciales de las MUs etiquetados por su número correspondiente (Fig. 14 (a)). Se pueden ver todos los MUP detectados y la forma de onda que tienen (Fig. 14 (b)) además de los patrones de disparo representados mediante círculos equiespaciados en el tiempo (Fig. 14(c)). Disponemos también de una ventana de ajuste de precisión (Fig. 14 (d)), donde podremos realizar un ajuste manual con el objetivo de resolver las superposiciones.

A partir de estos datos, es necesario revisar toda la señal EMG de principio a fin con el objetivo de mejorar los resultados mediante un proceso manual. Los errores más comunes que precisan de un ajuste manual son los siguientes:

- **MUPs no detectados:**

En la señal EMG, nos encontraremos con formas de onda que no han sido etiquetadas y que se repiten cada cierto tiempo. Para detectar estas formas de onda que no han sido etiquetadas debemos observar el residuo de la señal, la señal amarilla de la Fig. 14 (a). El residuo, es la señal de error entre los MUPs detectados y la señal real, dicho de otra forma, todo aquello que ha quedado sin resolver en la señal EMG.

En la Fig. 15, podemos observar un ejemplo en el que ocurre este error. Al final de la señal EMG (marcado en el rectángulo verde) se muestra un potencial que no ha sido detectado. Este potencial es muy similar al potencial número 5, con lo que debemos etiquetarlo manualmente.

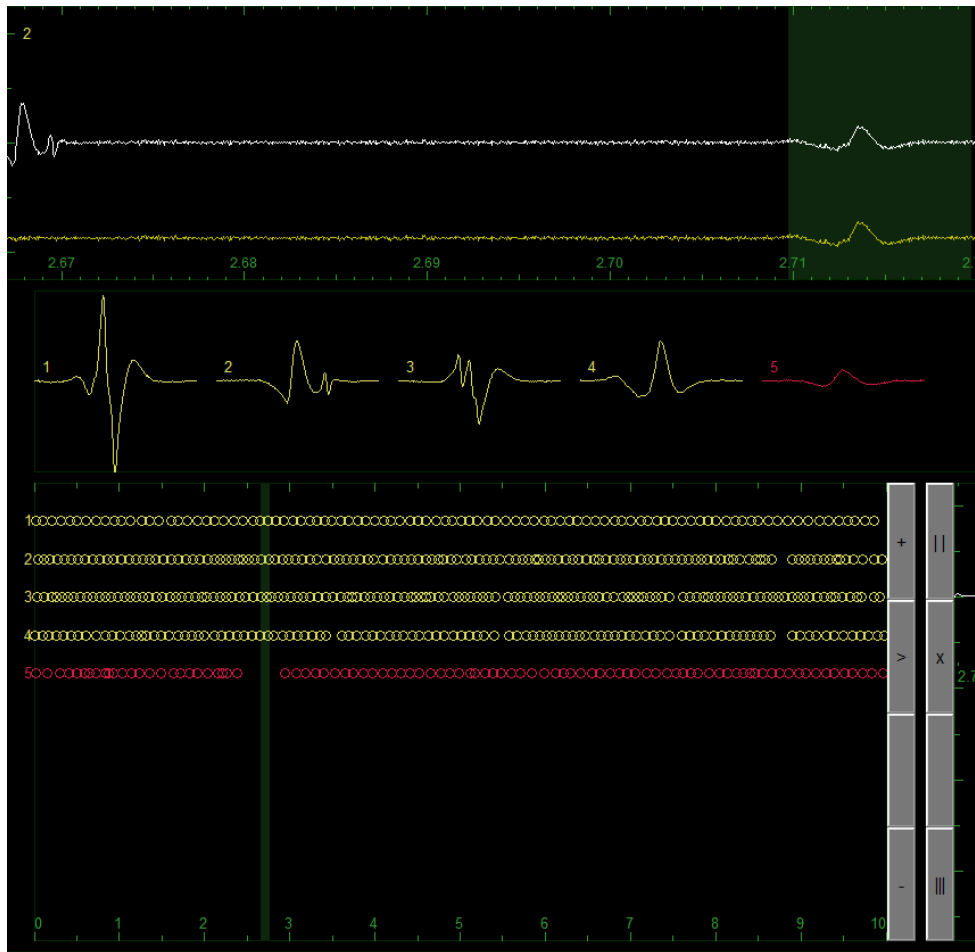


Figura 15. Ejemplo de MUP no detectado

- **Huecos en los instantes de disparo:**

En la ventana donde se muestran los patrones de disparo, se aprecian huecos. Estos huecos indican que el algoritmo no ha detectado el disparo cuando debería haberlo hecho. En la Fig. 15, el MUP no detectado corresponde con uno de esos huecos.

Para buscar MUPs que el algoritmo no ha podido detectar, los huecos nos servirán de guía.

- **MUPs superpuestos:**

Es posible que el algoritmo no haya podido resolver casos de superposición de MUPs. En el ejemplo de la Fig. 16, vemos un caso sin resolver que podría ocurrir con la superposición de los MUPs número 2 y 3.

Para resolver este tipo de casos contamos con la ayuda de la ventana de precisión (Fig. 17), donde realizamos un ajuste manual de la ubicación de los MUPs que en este caso se superponen.

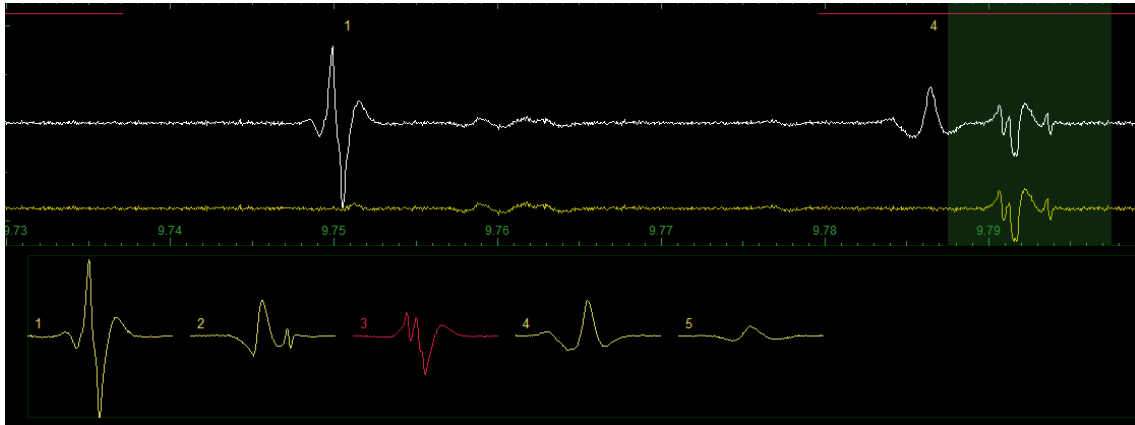


Figura 16. Ejemplo de MUPs superpuestos no detectados

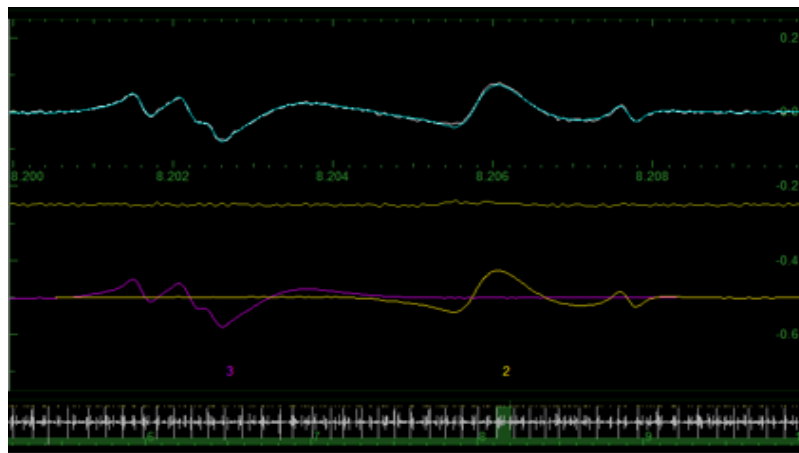


Figura 17. Ventana de ajuste de precisión

Limitaciones:

- **Efecto de la MVC**

Si la contracción muscular es alta disparan muchas unidades motoras, y la descomposición resulta más difícil. En la Fig. 18, podemos ver un caso de una MVC alta en la que se han detectado 9 MUPs. El ajuste manual para estos casos es muy complicado ya que hay muchos potenciales que son muy parecidos y el número de MUPs superpuestos aumenta mucho.

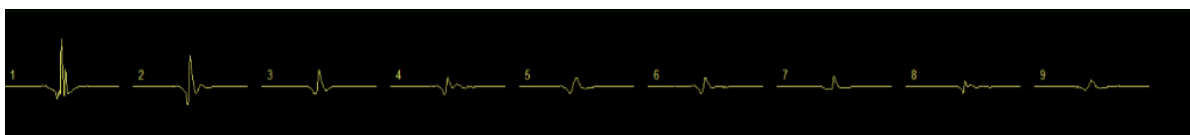


Figura 18. MUPs detectados para un MVC del 7%.

- **Ajuste manual:**

En los sistemas manuales se precisa de una inspección visual constante por parte del usuario, esto hace que sea probable llegar a conclusiones erróneas además de que requiere un tiempo excesivo para su evaluación.

3.2 Descomposición automática

Además de los sistemas de descomposición manuales se han desarrollado otros algoritmos que realizan todo el procedimiento de la descomposición de forma automática, como es el caso del algoritmo de Montreal.

Este algoritmo nos muestra como resultado los trenes de disparos de las unidades motoras que ha podido detectar. El tren de disparos obtenido se compara con los disparos reales que ya conocemos y evaluarán el éxito del procedimiento que se ha llevado para ello, mediante los siguientes parámetros:

- TP (Verdaderos positivos): Son los disparos que se han detectado correctamente.
- FP (Falsos positivos): Se trata de los instantes de disparo que el sistema ha detectado cuando no debería haberlo hecho por el hecho de que no han ocurrido. Estos casos resultan muy perjudiciales ya que lo único que nos aportan es ruido.
- FN (Falsos negativos): Se trata de disparos que el sistema no ha detectado cuando sí debería de haberlo hecho, porque han ocurrido. Estos casos no son tan perjudiciales ya que únicamente supone un menor número de disparos obtenidos.
- A (Accuracy): Será el porcentaje de disparos que se han conseguido identificar correctamente.

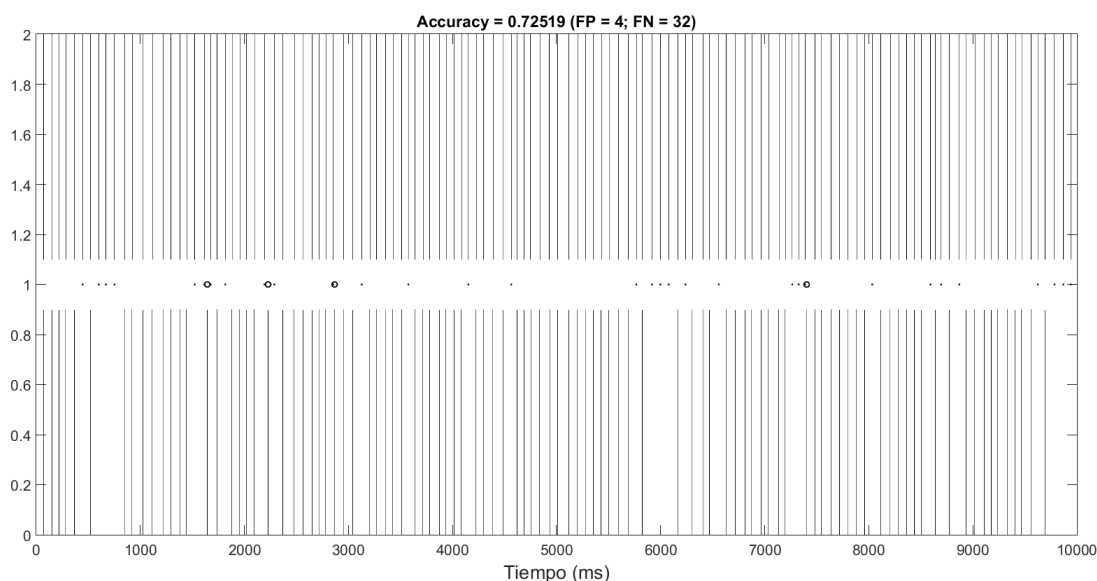


Figura 19. Representación de disparos reales (parte superior) frente disparos detectados (parte inferior) junto a los parámetros de FP (círculo) y FN (punto).

Capítulo 4. Evaluación de los sistemas de clasificación automática

En este capítulo, se definen y describen los métodos de evaluación de los sistemas de clasificación automática. Además, se describe el funcionamiento del sistema de descomposición automática implementado en este proyecto además de su evaluación. Los métodos para la realización de este sistema están basados en el proyecto [7]. A la vista de los resultados de la evaluación, se ha modificado el algoritmo de segmentación de [7] para corregir los errores detectados.

4.1 Preprocesado

En una señal EMG, nos encontramos con ruido de fondo y derivas de baja frecuencia en la línea de base además de ruido instrumental de alta frecuencia. En primer lugar, es necesario aplicar un filtro apropiado a nuestra señal EMG, con el objetivo de eliminar el ruido asociado a las componentes de baja y alta frecuencia, ya que no aportan información fisiológica.

Desde el punto de vista de la descomposición, nos interesa que los MUPs de las distintas unidades motoras se diferencien entre sí para que en el proceso de clasificación puedan ser asignadas a sus correspondientes clases. Teniendo esto en cuenta, se ha realizado un filtrado paso banda tipo Butterworth de segundo orden, entre 1KHz y 2KHz. Este filtro se ha logrado empleando un filtro paso alto seguido de un filtro paso bajo. Y los criterios para elegir los parámetros del filtrado paso banda se han extraído de [7]. Como se puede apreciar en la Fig. 20 (b), el filtrado implica dos efectos en nuestra señal EMG de la Fig. 20 (a): la disminución en amplitud y la compresión temporal de los MUPs.

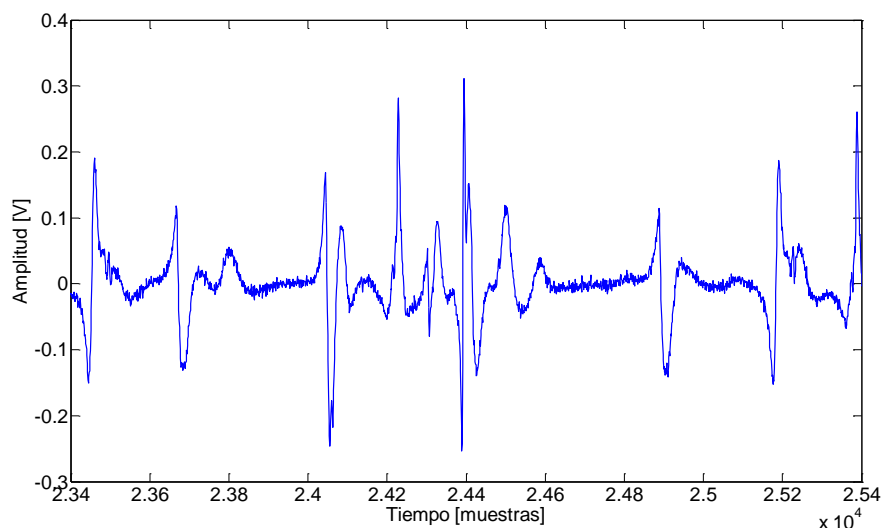


Figura 20. a) Preprocesado. Señal EMG original.

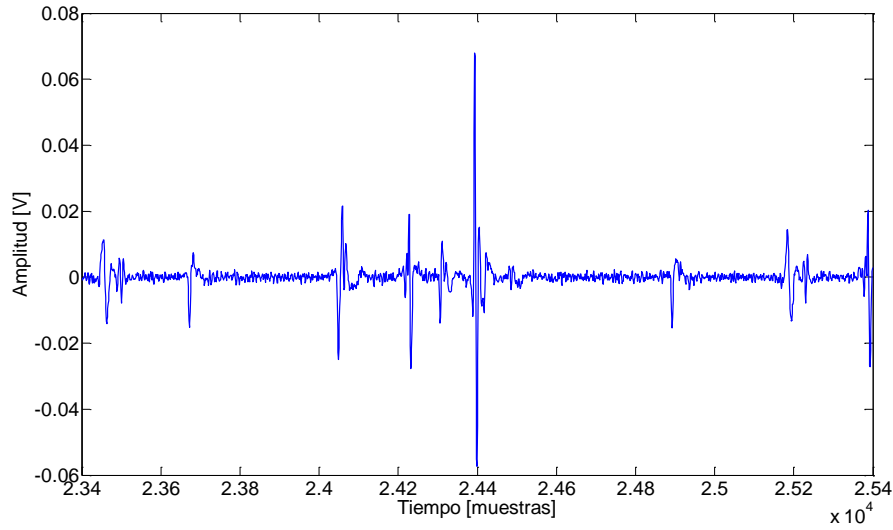


Figura 20. b) Preprocesado. Filtrado de la señal EMG.

La disminución en amplitud implica que desaparezcan los MUPs de las unidades motoras que se encuentran en territorios lejanos respecto al electrodo debido a su baja amplitud, lo cual nos facilita la posterior detección para extraer los segmentos. En los demás casos, la disminución de amplitud no supone mayor problema ya que los MUPs de una misma MU siguen pareciéndose y se diferencian respecto de otras MUs.

El efecto de la compresión temporal reduce los solapamientos entre MUPs cercanos en el tiempo, lo cual resulta favorable para la clasificación. Estos dos efectos hacen que cambie ligeramente la forma de los potenciales, sin embargo siguen siendo distinguibles entre sí.

4.2 Segmentación

Una vez tengamos la señal filtrada, el objetivo es extraer los segmentos que contengan un MUP o una superposición de MUPs de las unidades motoras.

4.2.1 Detección

La detección es el primer paso para encontrar los segmentos de la señal. Para llevarla a cabo, se define un umbral de detección y se almacenan los cruces que hace la señal filtrada EMG por este umbral con pendiente positiva (Fig. 21).

El umbral debe elegirse de forma que se puedan diferenciar por encima del umbral los MUPs más significativos que puedan ser parte del ruido de fondo. Dicho esto, el umbral que se ha empleado en este proyecto es el umbral de Quiroga [8]. Su valor se obtiene mediante la siguiente fórmula:

$$Umbral = 4\sigma_n; \quad \sigma_n = mediana \left\{ \frac{|x_{fil}|}{0,6745} \right\} \quad (1)$$

donde x_{fil} es la señal filtrada y σ_n es considerada una buena estimación de la desviación estándar del ruido de fondo.

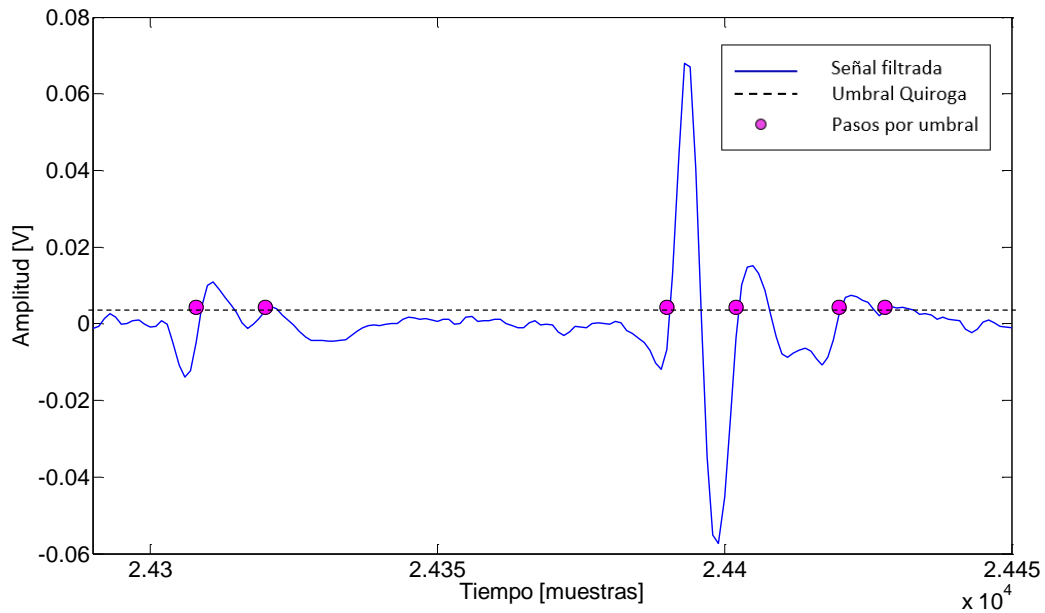


Figura 21. Cruces de la señal filtrada EMG con el umbral de Quiroga.

4.2.2 Búsqueda de los máximos

A partir de los instantes de los cruces con el umbral, se busca el máximo entre un cruce y el próximo. En la Fig. 22, se puede ver el resultado de esta búsqueda; el primer máximo se busca entre el instante del cruce (1) y el cruce (2), el siguiente máximo es el que se encuentra entre el cruce (2) y el (3) y así sucesivamente. El último máximo se encuentra en el rango desde el último cruce hasta el final de la señal.

Este es el primer cambio introducido respecto de la versión de [7] en la que se buscaba el máximo en una ventana prefijada.

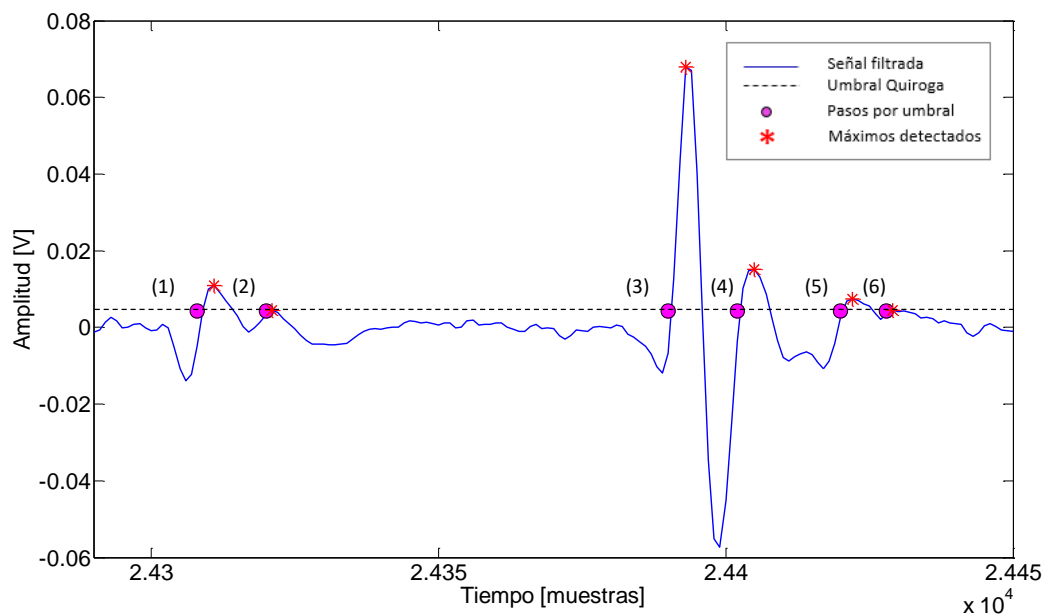


Figura 22. Máximos obtenidos a partir de los instantes de los cruces por umbral

4.2.3 Exclusión de máximos

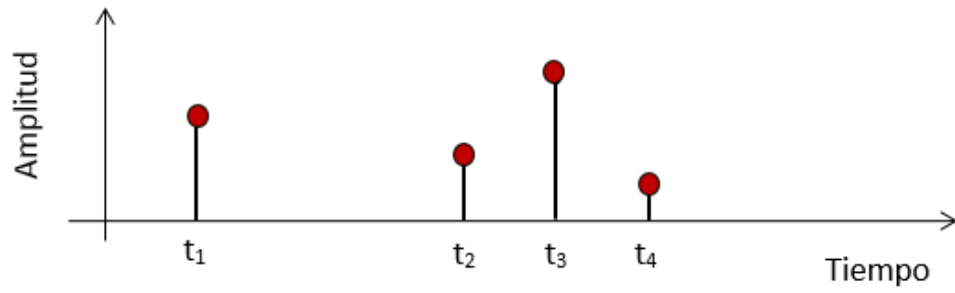
Ya tenemos detectados todos los máximos de la señal. Ahora, necesitamos quedarnos solamente con el pico máximo que represente a cada posible MUP o superposición de MUPs para después extraer los segmentos centrados en el instante de dicho máximo. Como no interesa tomar picos excesivamente cercanos (puesto que pueden pertenecer al mismo MUP), el objetivo es filtrarlos y quedarnos con un único pico que represente a su MUP correspondiente.

Para ello, disponemos de un parámetro libre del algoritmo que define la ventana de exclusión. En un primer momento, tal y como se propone en [7], definiremos una ventana de 4 ms (80 muestras a 20 KHz). Más adelante decidiremos cuál es la ventana óptima. Una vez definida, fijaremos como primer candidato a ser uno de los máximos finales, el primer máximo de nuestra señal. A partir de aquí, durante el proceso de búsqueda nos encontraremos con 2 casos que deciden si el máximo que estamos evaluando pasa a ser el máximo final o no.

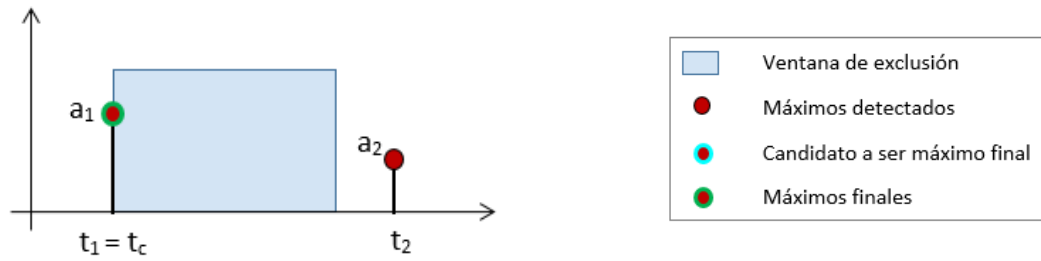
En la Fig. 24, se muestra mediante un diagrama de flujo el procedimiento que se lleva a cabo para la búsqueda de los máximos finales.

- Lo primero que se comprueba es la distancia que hay entre el candidato y el próximo máximo a este. Si esta distancia es mayor que la ventana de exclusión, (Fig. 23 (b), *caso 1*), significa que el candidato es un máximo que se encuentra lo suficientemente lejos de su próximo como para considerar que nos encontramos ante el pico máximo que representa a un MUP. Con lo cual, este candidato pasará a ser uno de los máximos finales que buscamos.
- En cambio, si la distancia es menor que la ventana, (Fig. 23 (c) y (d), *caso 2*), significa que los dos máximos pertenecen al mismo MUP, con lo cual, lo siguiente que se comprueba es la diferencia de amplitudes entre ambos máximos para decidir si el candidato debe ser otro o sigue siendo el mismo: si la amplitud del candidato actual es mayor que su próximo máximo (Fig. 23 (d), *caso 2.2*), el candidato seguirá siendo el mismo hasta que se encuentre un máximo de mayor amplitud o un máximo a mayor distancia que la ventana de exclusión. Pero si la amplitud de dicho candidato es menor que su próximo máximo (Fig. 23 (c), *caso 2.1*), el candidato pasará a ser el que tiene mayor amplitud.

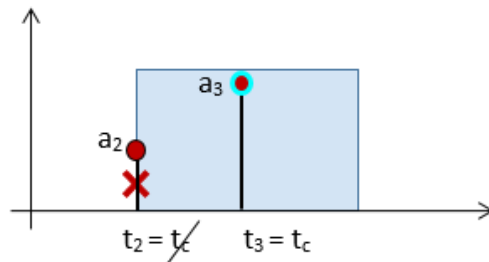
Resumiendo, si la distancia entre dos máximos es menor que la ventana, el candidato siempre será el de mayor amplitud y cuando la distancia entre el candidato y un máximo sea mayor que la ventana, es cuando este candidato pasa a ser uno de los máximos finales. Este procedimiento que hemos seguido para la exclusión de máximos también es diferente al de [7], el cual genera más FNs como veremos más adelante.



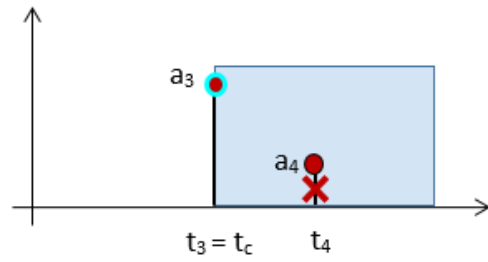
a) Máximos obtenidos a partir de los instantes de los cruces por umbral



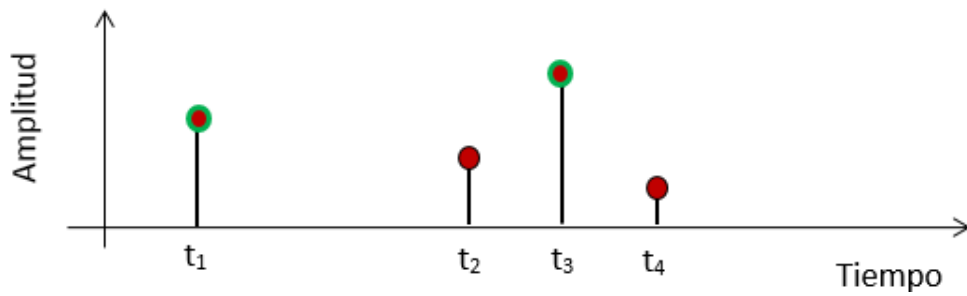
b) **Caso 1:** $t_k - t_c > \text{win_exclude}$



c) **Caso 2.1:** $a_k - a_c > 0$



d) **Caso 2.2:** $a_k - a_c \leq 0$



e) Máximos finales obtenidos.

Figura 23. Representación del proceso de exclusión de picos cercanos.

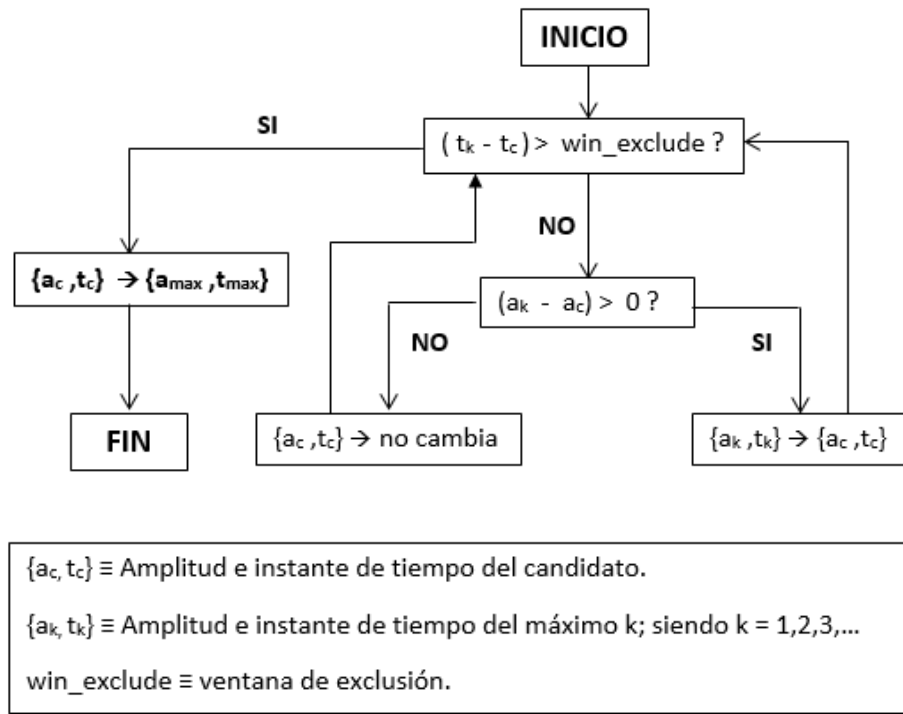


Figura 24. Diagrama de flujo del proceso de exclusión de picos cercanos.

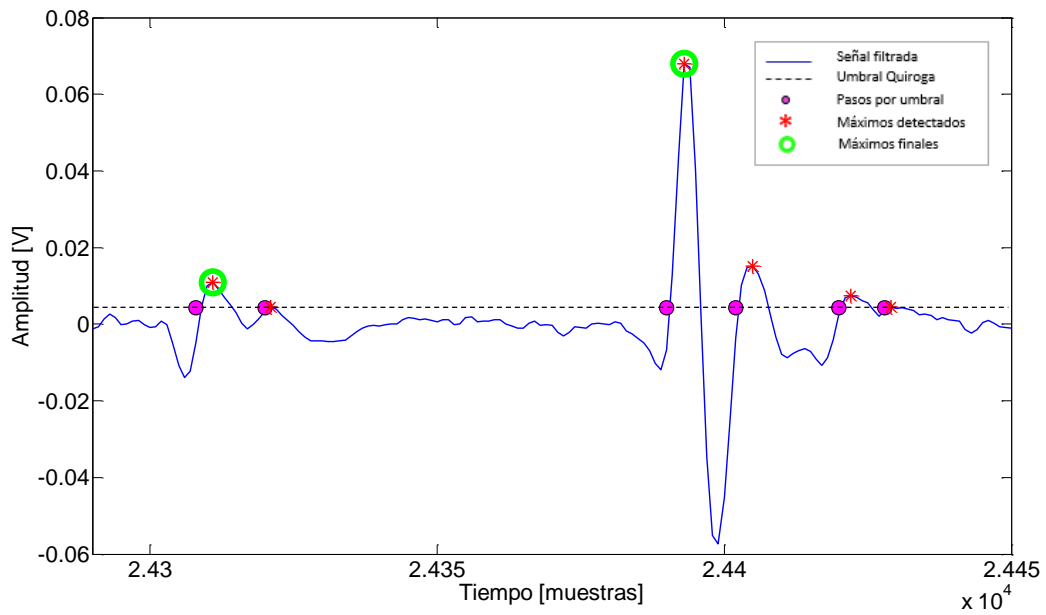


Figura 25. Máximos finales de los segmentos

4.2.4 Enventanado de los segmentos

Una vez que tengamos los máximos finales (Fig. 25), los cuales representan el máximo de cada MUP o superposición de MUPs, extraeremos los segmentos en torno a los picos mediante una ventana de adquisición simétrica de 4 ms (39 muestras por delante y 40 muestras por detrás del máximo) ya que la duración de un MUP es de 4 ms aproximadamente (Fig. 26).

De esta manera quedan almacenados tantos segmentos como máximos finales tengamos en la señal.

4.2.5 Instantes de disparo

Los instantes de disparo son precisamente los instantes de los máximos sobre los que se han centrado los segmentos. Estos se almacenan en un vector y más adelante, tras el proceso de clasificación, veremos a qué unidad motora corresponde cada disparo.

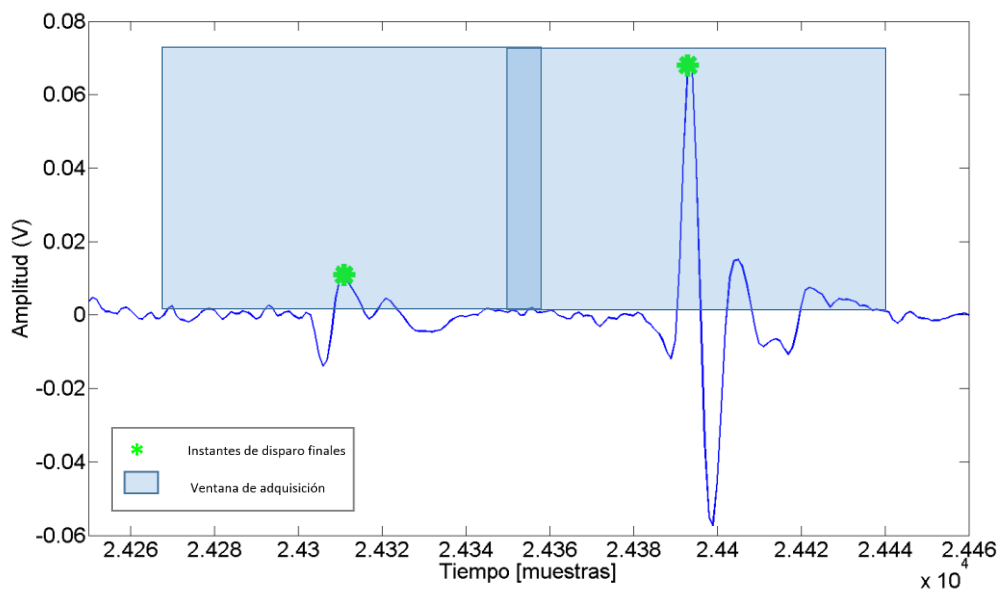


Figura 26. Instantes finales de disparo y ventana de adquisición.

4.3 Evaluación de la segmentación

En este apartado, se procede a evaluar el proceso de segmentación desarrollado con el fin de determinar su rendimiento. Para evaluar los resultados se han utilizado algunas figuras de mérito como la precisión (*accuracy*), la tasa de disparos no detectados (falsos negativos) y la tasa de falsos positivos.

4.3.1 Determinación del rendimiento de la segmentación

Conociendo los trenes de disparos reales podremos compararlos con los obtenidos en la segmentación y calcular así las figuras de mérito que nos permitirán conocer la calidad del algoritmo de segmentación. Para realizar correctamente esta comparación, es importante seleccionar entre los disparos reales aquellos que cumplan las mismas condiciones que los obtenidos por el algoritmo de segmentación. Para ello, se utiliza la siguiente función:

```
[T_ref]=train_ref(X,muscle,electrode,decomp,mup,T)
```

Parámetros de entrada:

- *X*: señal EMG original.
- *muscle*: estructura de datos que contiene la información del modelo del músculo.
- *electrode*: estructura de datos que contiene la información del electrodo.
- *decomp*: estructura de datos que contiene la información de la descomposición.
- *mup*: estructura *cell-array* que contiene los MUPs ideales de cada MU, sin interferencias de otras unidades motoras.
- *T*: estructura *cell-array* que contiene los instantes de disparo ideales (en ms) de las unidades motoras del músculo.

Parámetros de salida:

- *T_ref*: vector que contiene de forma ordenada los instantes de disparo de referencia (en ms) para la segmentación.

Mediante la siguiente función se obtienen los trenes de disparo de referencia siguiendo el siguiente procedimiento:

1. Filtrar todos los MUP de las MUs activas mediante el mismo filtro que se ha empleado en el pre-procesado de la segmentación.
2. Obtener los índices de aquellas MUP cuyo pico máximo supere el umbral impuesto en el algoritmo de segmentación.
3. Quedarnos con los instantes *T* correspondientes a las MUs obtenidas en el punto anterior.
4. Sumar el tiempo de retardo de propagación, Δt , a dichos instantes obteniendo los instantes de referencia *T_ref*.

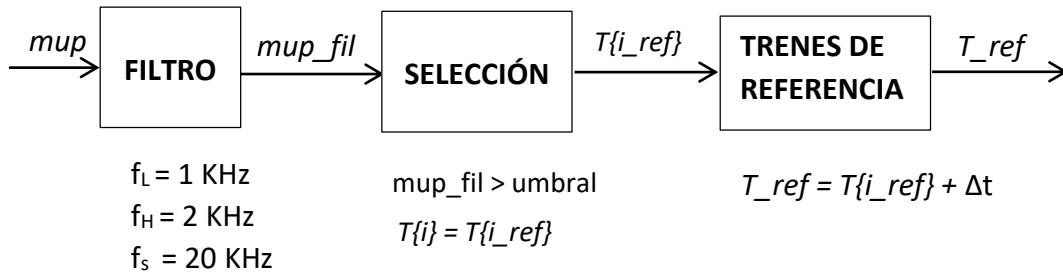


Figura 27. Diagrama de bloques para la obtención de los trenes de referencia.

Como se ha comentado anteriormente, los segmentos representan a un MUP o superposición de MUPs centrados en los instantes de disparo que evaluaremos. Debemos tener en cuenta que dichos MUPs se han obtenido sobre una señal filtrada, con lo cual, el primer paso es filtrar en nuestro entorno simulado, todos los MUPs de las MUs que estén activas. Una vez filtradas debemos quedarnos con aquellos MUPs cuyo pico máximo haya superado el umbral impuesto en la segmentación, el umbral de Quiroga en este caso. De esta forma, ya conocemos todos los segmentos que el algoritmo debería haber detectado.

Por último, un aspecto a tener en cuenta es el retardo de propagación. Los disparos reales del entorno de simulación son los instantes en los que el potencial se inicia en la zona de inervación, mientras que los disparos de la segmentación se detectan cuando los potenciales llegan al electrodo. Por lo tanto, para que la comparación sea lo más exacta posible, sumaremos el retardo de propagación a los disparos seleccionados. Dicho retardo, se calcula conociendo la posición del electrodo y la media de la zona de inervación además de las velocidades de conducción de cada una de las MUs que se encuentran activas (Fig. 28).

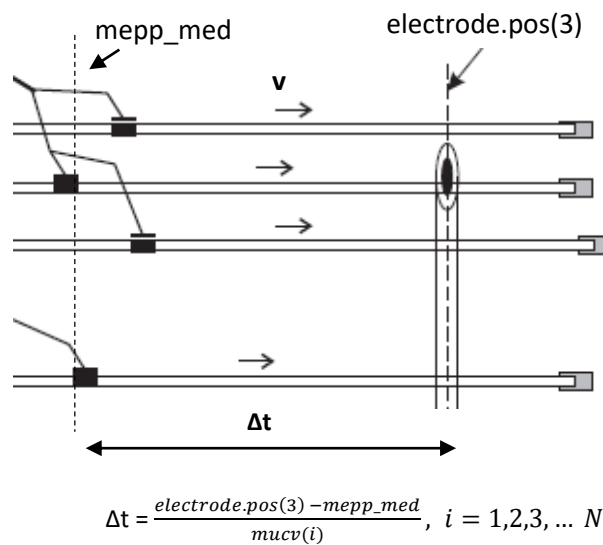


Figura 28. Representación esquemática del cálculo del retardo de propagación.

Una vez alineados, se calculan los parámetros TP, FP y FN. Según el caso que nos ocupe, dichos parámetros se definen de distintas maneras.

Un segmento que contiene una superposición de MUPs, en el algoritmo de segmentación se considera como un único MUP, un disparo, cuando en realidad se han producido varios en ese segmento. Como en este caso se está evaluando la capacidad de detectar segmentos, todos aquellos disparos que forman parte de una superposición de MUPs se asocian a un único disparo; el que representa el segmento. Como se muestra en la Fig. 29, este disparo será un verdadero positivo. Para determinar qué disparos son los que forman parte de la superposición se define un margen de tiempo de 4 ms, duración aproximada de un MUP. Por lo tanto, el disparo detectado en la segmentación se asocia a todos los disparos dentro de dicho margen.

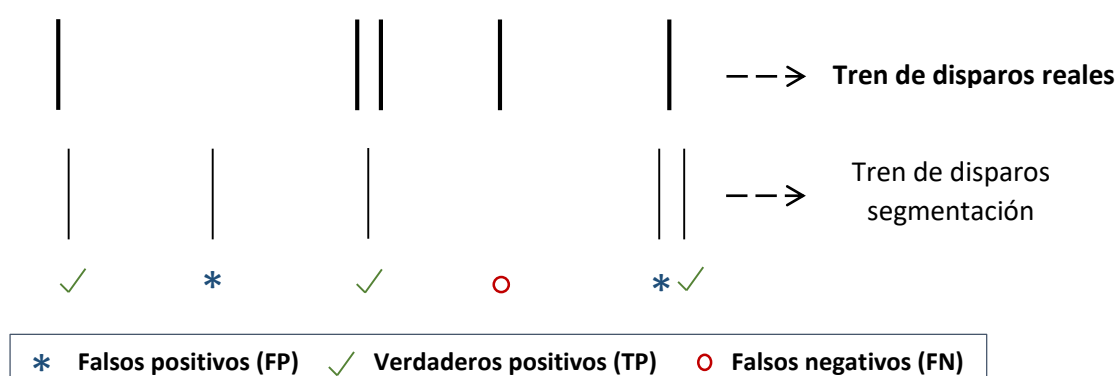


Figura 29. Diagrama de definición de los FP, FN y TP en la evaluación de la segmentación.

Para cuantificar el rendimiento del algoritmo de segmentación implementado, se realizará un experimento donde se generan 100 músculos de distinto tamaño. Para cada realización se obtiene: una señal EMG registrada por el electrodo de aguja, los instantes de disparo obtenidos en el proceso de segmentación, los instantes de disparo reales seleccionados, y los parámetros de *accuracy*, FP, FN y TP.

Estas realizaciones se calcularán variando los siguientes parámetros:

- **MVC:** nivel de contracción voluntaria. Cuanto mayor sea el nivel de MVC, mayor es el número de MUs reclutadas y la frecuencia de disparo. En este caso, la variación del MVC se hará del 2 al 40% con incrementos del 1%.
- **win_exclude:** ventana de exclusión de máximos. Este parámetro se varía con el fin de determinar la ventana óptima. En este caso, la variación se realiza desde una ventana de 5 muestras a una de 180 con incrementos de 5 muestras. Este experimento se realizará 4 veces fijando un MVC del 3, 6, 12 y 24%, para ver la influencia del MVC junto a la variación de la ventana.

4.3.2 Mejoras del algoritmo de segmentación

El algoritmo desarrollado en el apartado 4.2, es una versión mejorada del algoritmo implementado en [7]. Para determinar el grado de mejora del algoritmo, se han comparado las figuras de mérito obtenidas en ambos casos después de realizar una simulación donde variamos el MVC con la ventana de exclusión propuesta en [7], una ventana de 80 muestras.

En la Fig. 30, se muestran las figuras de mérito de ambos casos a modo de diagrama de caja (*box plot*), donde se observa la distribución de 100 músculos diferentes. De los resultados obtenidos, la exactitud (*accuracy*) es el parámetro que más nos interesa, ya que nos indica cómo de acertado ha sido el proceso de segmentación. Como podemos ver en la Fig. 30 (a), el nivel de *accuracy* de [7] decrece conforme aumenta el MVC hasta llegar a un 80% aproximadamente en el nivel más alto de MVC. Mientras que en el algoritmo mejorado (Fig. 30 (b)) el nivel de *accuracy* se mantiene más estable y decrece de forma más lenta teniendo un 90% de *accuracy* para el nivel de MVC más alto.

La gran diferencia entre los dos algoritmos se encuentra en el porcentaje de los falsos negativos detectados, razón por la cual se ha logrado mejorar el nivel de *accuracy*. Como vemos en la Fig. 30 (c), el algoritmo de [7] presenta una curva creciente a medida que aumenta el nivel de MVC al igual que el algoritmo mejorado (Fig.30 (d)) pero éste último consigue detectar un porcentaje de FNs menor.

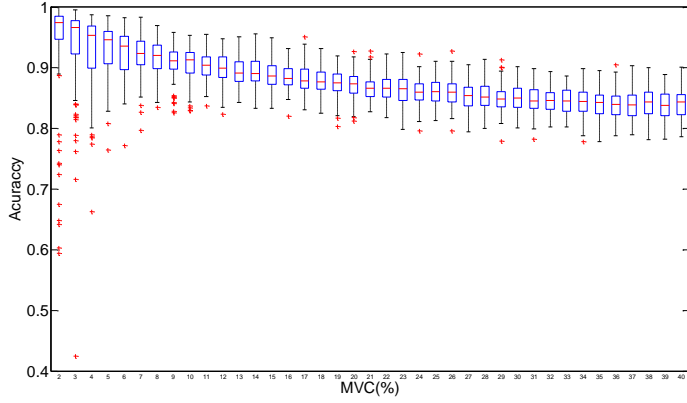
En cuanto a los FPs, el resultado es muy similar en ambos casos. La única diferencia que se observa entre ambos se encuentra en niveles de contracción bajos (entre el 2% y el 7% aproximadamente), donde el algoritmo [7] (Fig. 30 (e)) obtiene un mejor resultado con un porcentaje menor de FPs que en el algoritmo mejorado (Fig. 30 (f)). Como podemos ver esta diferencia es muy pequeña por lo que se puede considerar despreciable.

A partir de estos resultados, con el fin de poder apreciar de una forma más precisa las mejoras que se han logrado, se ha realizado una comparación mediante las curvas del primer y tercer percentil de los *box plot* de ambos algoritmos. Y a partir de la mediana se ha obtenido la curva de ganancia (Fig. 31).

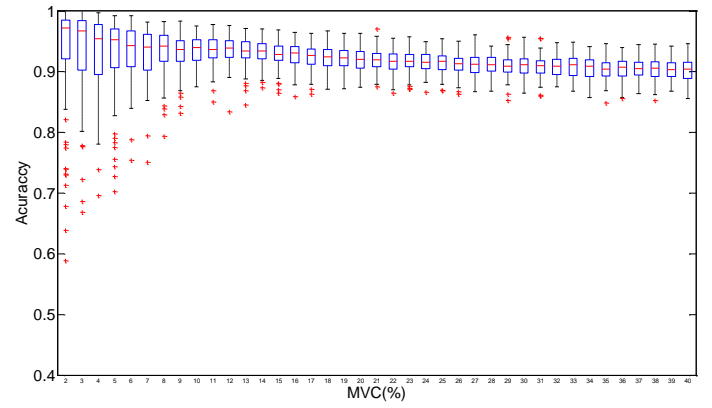
En la Fig. 31 (b), se observa que la ganancia incrementa hasta llegar a un 7% para el nivel de contracción más alto. Este incremento en la ganancia, hace que la caída del *accuracy* respecto del MVC en nuestro algoritmo sea más lento (Fig. 31 (a)). En nuestro algoritmo el *accuracy* decrece hasta llegar al 90% para niveles de MVC del 40%, mientras que en el algoritmo de [7], ese mismo porcentaje de *accuracy*, 90%, decrece a partir del MVC del 10%. Por lo tanto, se ha conseguido mejorar la segmentación de forma más notable en niveles de contracción altos.

Como ya hemos visto, esto es debido a un menor número de falsos negativos detectados en nuestro algoritmo. Como vemos en la Fig. 31 (c), la curva de ganancia del porcentaje de los falsos negativos detectados, es prácticamente la misma curva que la ganancia obtenida en el *accuracy*.

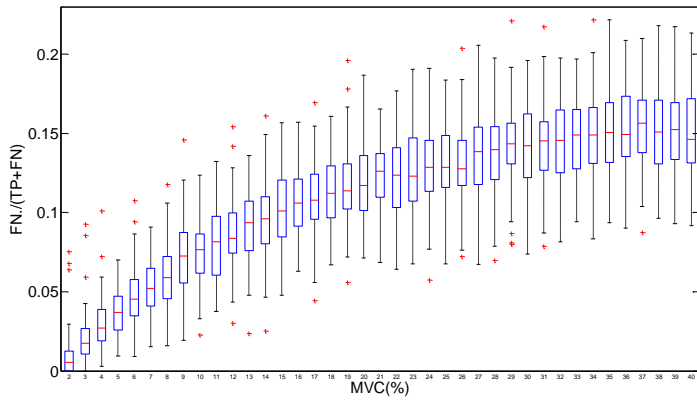
Por último, la curva de ganancia para el caso de los FPs (Fig.31 (f)) se mantiene prácticamente constante a partir de un MVC del 10%, por lo tanto respecto a los FPs no se ha conseguido una mejora apreciable. Por debajo de este nivel la ganancia sufre una caída de aproximadamente un 1%, lo cual consideramos despreciable.



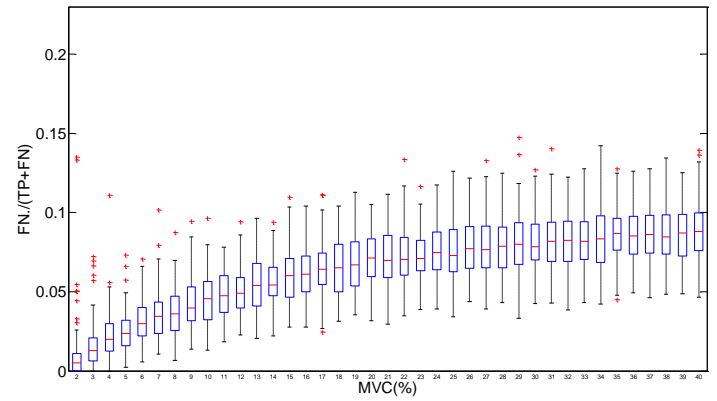
a)



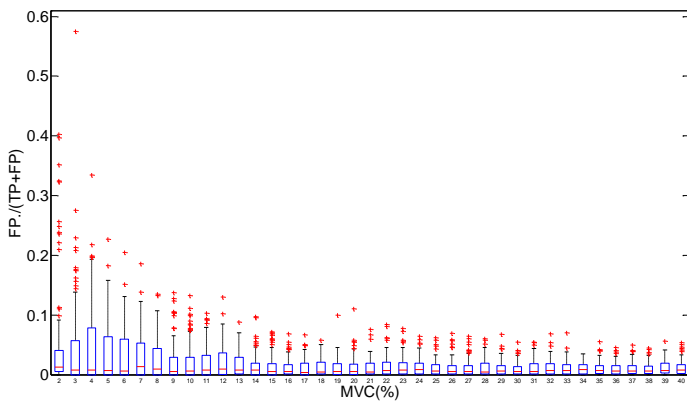
b)



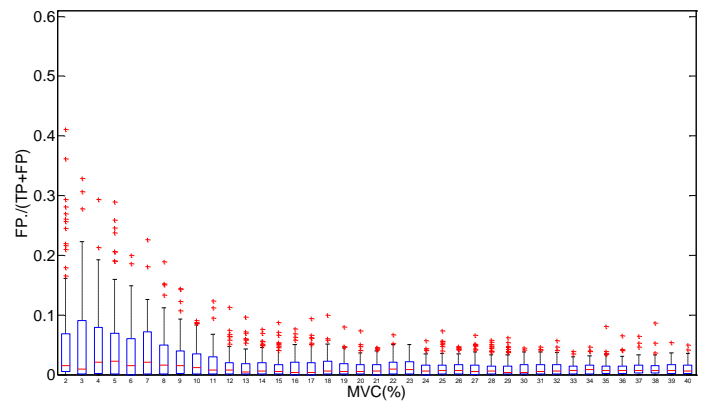
c)



d)

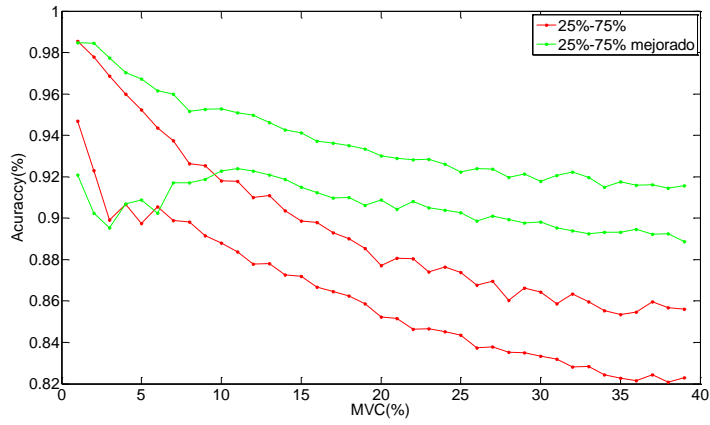


e)

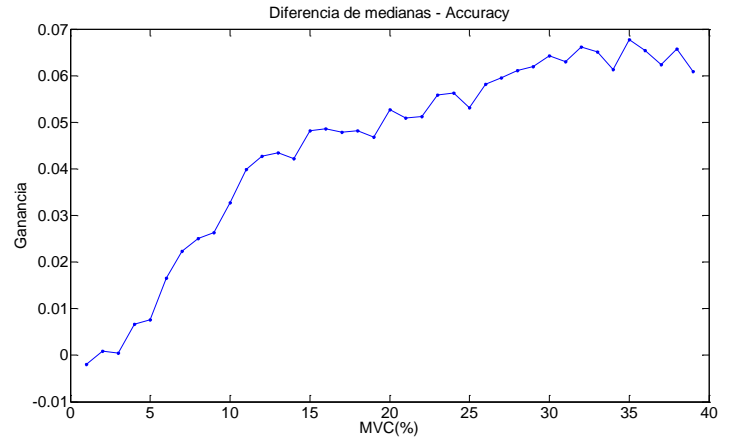


f)

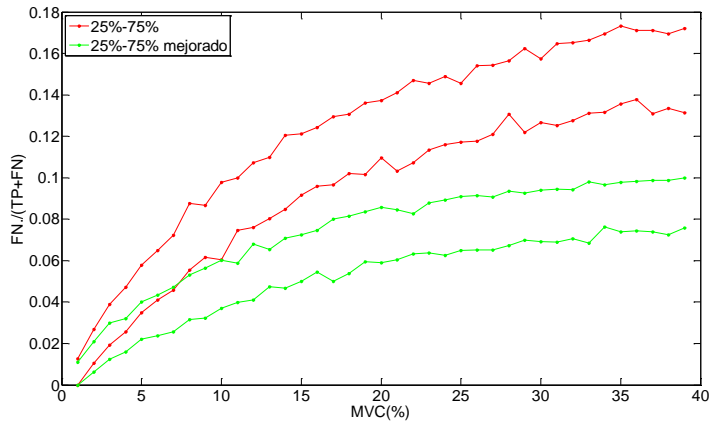
Figura 30. Comparación del algoritmo de [7] y su versión mejorada. a) c) y e) son las figuras de mérito correspondientes al algoritmo [7] y b) d) y f) los correspondientes a su versión mejorada.



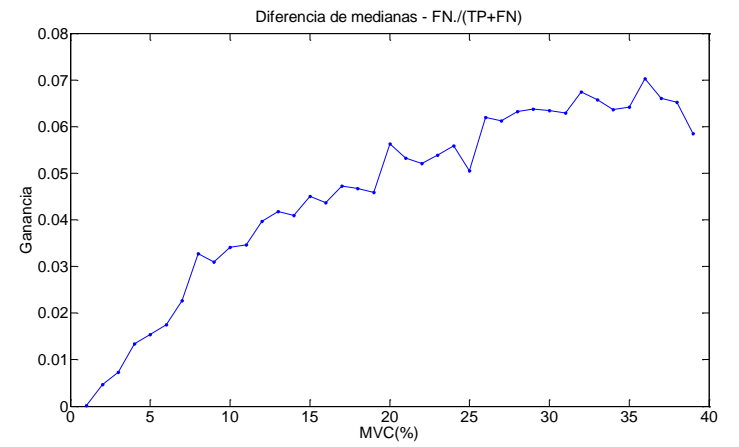
a)



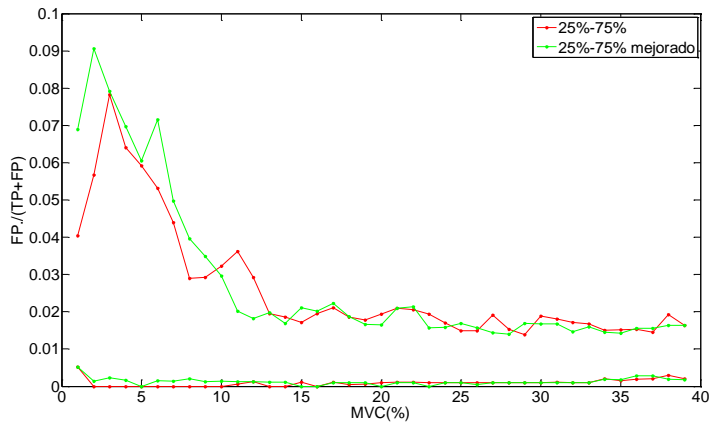
b)



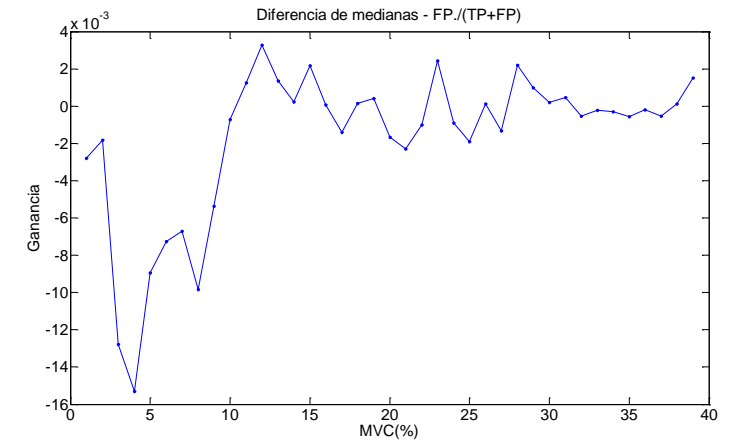
c)



d)



e)



f)

Figura 31. Comparación del algoritmo de [7] y su versión mejorada en el rango del percentil 25% al 75%. a) Accuracy. b) Ganancia accuracy. c) Falsos negativos. d) Ganancia falsos negativos. e) Falsos positivos. d) Ganancia falsos positivos.

Errores en el algoritmo de [7]:

La razón por la que en el algoritmo [7] se detecten más falsos negativos ha sido el método implementado para la búsqueda de los máximos a partir de los cruces por umbral, y también el implementado para la exclusión de dichos máximos:

- **Búsqueda de máximos**

Para encontrar los máximos de la señal, a partir de cada uno de los cruces por umbral se utiliza una ventana de búsqueda fijada en 4ms (80 muestras a 20kHz de frecuencia de muestreo) y en ese intervalo se busca el máximo absoluto. El empleo de esta ventana de búsqueda no recoge algunos máximos entre cruce y cruce, lo que conlleva a descartarlos directamente para seleccionarlos como máximos que representen a un segmento y por ello generar un posible falso negativo. En la Fig. 32, podemos ver un ejemplo de cómo la ventana de búsqueda es incapaz de detectar máximos entre cruce y cruce en este caso en dos ocasiones. Además se encadenan repeticiones de máximos, como ocurre en este caso, ya que la primera ventana recoge el mismo máximo que la segunda y la tercera ventana.

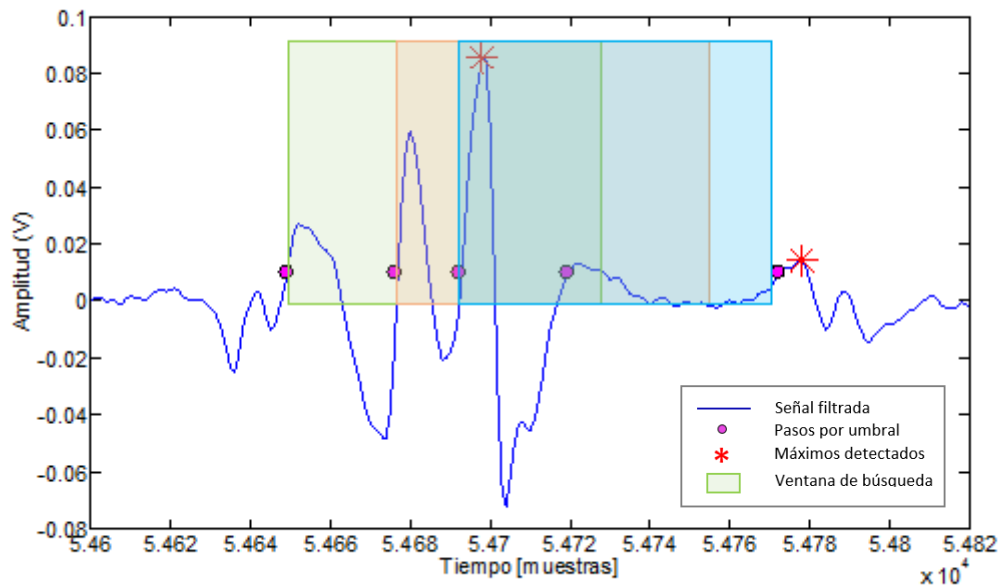


Figura 32. Ventana de búsqueda con los cruces por umbral de [7].

- **Exclusión de máximos**

Para seleccionar los máximos que representen a los segmentos que contienen los MUP, se ha utilizado la misma ventana de exclusión que en el algoritmo mejorado pero los máximos se han seleccionado teniendo en cuenta únicamente la primera condición (Caso 1 del apartado 4.2.3). Por ello, ocurren casos como el de la Fig. 33 donde se ha excluido un máximo que claramente representa un MUP.

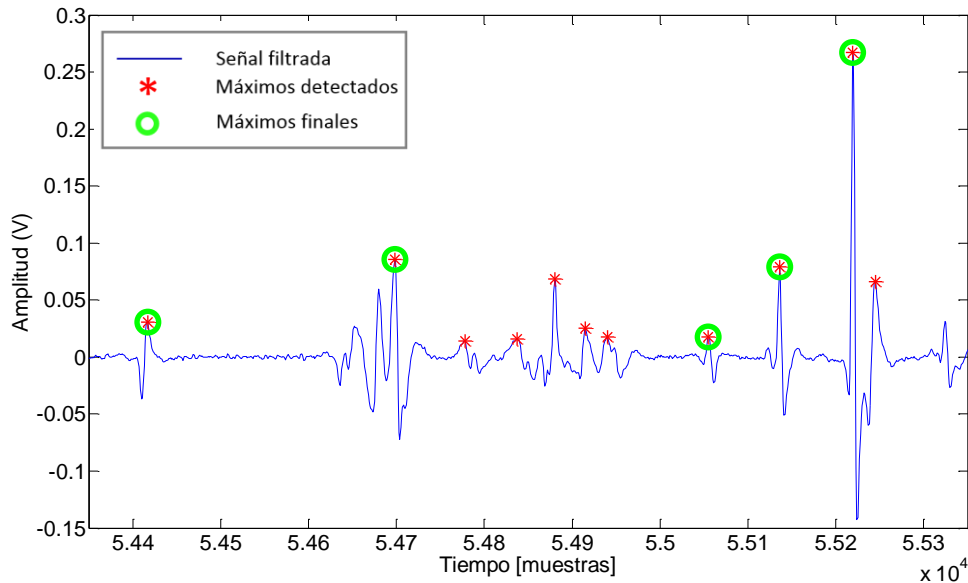


Figura 33. Errores en la búsqueda y exclusión de máximos de [7].

4.3.3 Variación del tamaño de la ventana de exclusión

En un primer momento se ha definido una ventana de exclusión de 4ms (80 muestras a 20kHz de frecuencia de muestreo), para asegurarnos de que ésta es una ventana óptima analizaremos los resultados obtenidos al variar el tamaño de la ventana.

Si observamos los resultados obtenidos de *accuracy* (Fig. 35), vemos que el porcentaje más alto se obtiene a partir de una ventana de 45 muestras. Por lo que el límite inferior del rango de tamaños de ventana óptimas es de 45 muestras. A partir de ahí conforme aumenta el tamaño de la ventana, el nivel de *accuracy* decrece y cuanto mayor es el MVC más rápido es este decremento. De forma que el límite superior lo definiremos como aquella ventana que no baje al menos de un 90% de *accuracy* tanto para niveles bajos como altos de MVC. Ya que la caída de *accuracy* es mayor para MVCs altos, el límite superior la obtendremos a partir de la Fig. 35 (d), una ventana de 80 muestras. Por lo tanto, el tamaño de la ventana óptima se encuentra en el rango de 45 a 80 muestras. Según los resultados de FNs y FPs obtenidos definiremos la ventana óptima definitiva dentro de dicho rango.

Observando los resultados de los falsos negativos (Fig. 36), cuanto mayor sea la ventana más máximos excluimos, lo que supone un mayor número de falsos negativos. Es decir, entre todos los máximos que se detectan, a la hora de decidir cuáles de ellos son los que representan a los segmentos que contienen los MUPs, la exclusión es mayor conforme aumente la ventana y esto supone un mayor porcentaje de falsos negativos en el resultado final. Por lo que si buscamos el menor número de FNs posible, la ventana óptima sería la más pequeña.

Sin embargo, los resultados de los falsos positivos (Fig. 37) nos indican que cuanto menor sea la ventana mayor será el porcentaje de falsos positivos. Esto está

relacionado con el margen de tiempo definido para que los disparos que hayan ocurrido en ese intervalo (superposición de MUPs) se asocien a un único disparo para así evaluar la correcta detección de un segmento (Apartado 4.3.1).

En la Fig. 34, se muestra un ejemplo donde nos encontramos con dos casos en los cuales se tienen dos máximos separados el mismo margen de tiempo, 4 ms. Además, el primer máximo (máximo en t_1) sabemos de antemano que se trata de un FP, mientras que el segundo máximo (máximo en t_2) es un TP. Mediante este ejemplo trataremos de ver qué ocurre si aplicamos distintos tamaños de ventana.

- En el primer caso, se emplea una ventana de exclusión de 60 muestras lo cual hace que, de acuerdo al procedimiento que seguimos para la exclusión de máximos, seleccionemos como máximo final el máximo en t_1 . El máximo en t_2 al estar dentro del margen que define un segmento, queda asociado al máximo en t_1 . Al realizar la evaluación de dicho segmento con los trenes de referencia, obtendremos un FP.
- En el segundo caso, se emplea una ventana de exclusión mayor, una de 80 muestras. Y de acuerdo al procedimiento que seguimos para la exclusión de máximos, seleccionamos como máximo final el máximo en t_2 . El máximo en t_1 por estar dentro del margen quedará asociado al máximo en t_2 . Después de evaluar este segmento con los trenes de referencia el resultado será un TP.

Con lo cual, los FPs que se encuentran dentro de segmentos representados por un máximo que resulta ser un TP, se convierten también en TP. Por esta razón se detectan menos falsos positivos cuanto mayor sea la ventana.

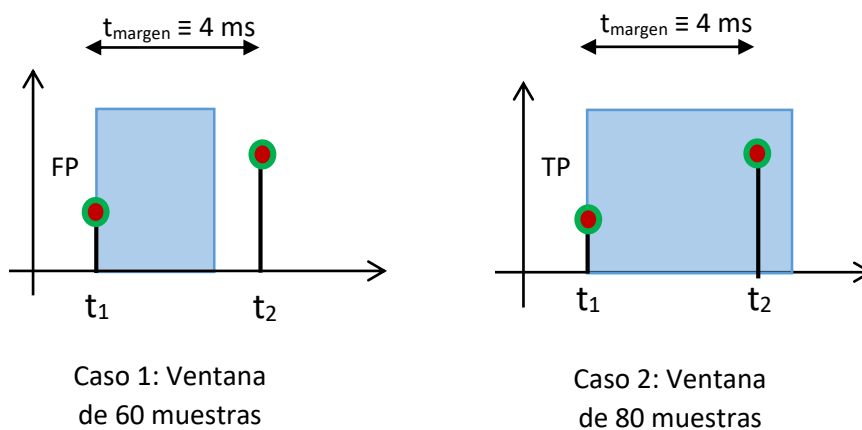


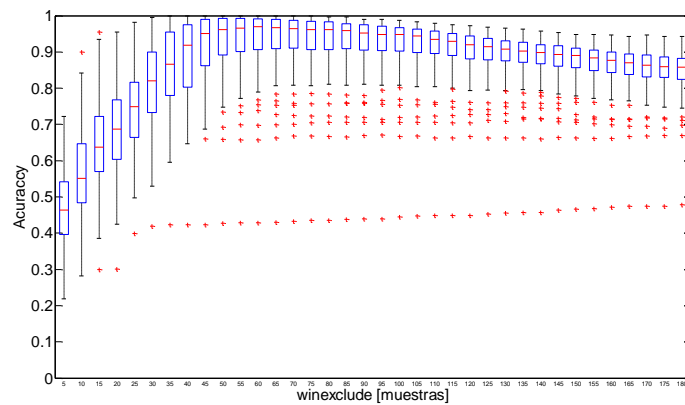
Figura 34. Representación de la relación entre la ventana de exclusión y los falsos positivos.

Teniendo en cuenta todos estos aspectos, si optamos por la ventana más pequeña (45 muestras) tendremos un menor número de FNs pero un mayor número de FPs, al contrario que si optamos por una ventana mayor. Los FPs son disparos que han sido detectados cuando en realidad no han ocurrido mientras que los falsos negativos suponen menos disparos detectados. Pensando en el proceso de clasificación que se

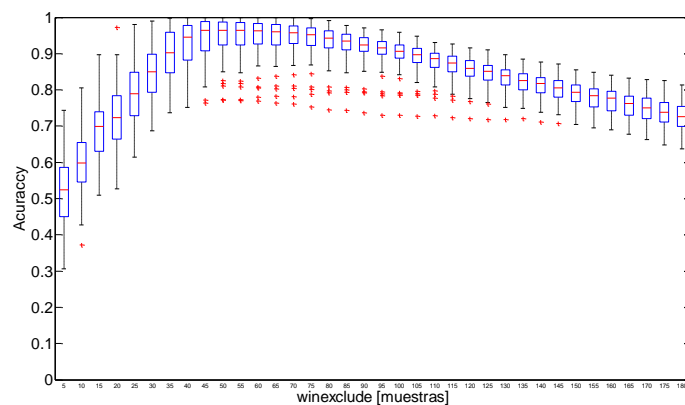
realiza tras la segmentación, los posibles falsos negativos supondrán segmentos que, al no haberse encontrado no podrán clasificarse, y su corrección sólo será posible recurriendo a un análisis del residuo de la señal descompuesta. Por otro lado, los posibles falsos positivos deberían ser descartados para el proceso de clasificación o, de lo contrario, generarían errores de clasificación en dicho proceso. Con lo cual, resulta difícil saber qué es lo que más nos podría perjudicar.

El tamaño de la ventana óptima por tanto, se decidirá según el nivel de *accuracy*. Y como ya hemos visto, la ventana de 45 muestras es la que mayor nivel de *accuracy* obtiene, por lo que dicha ventana será la óptima.

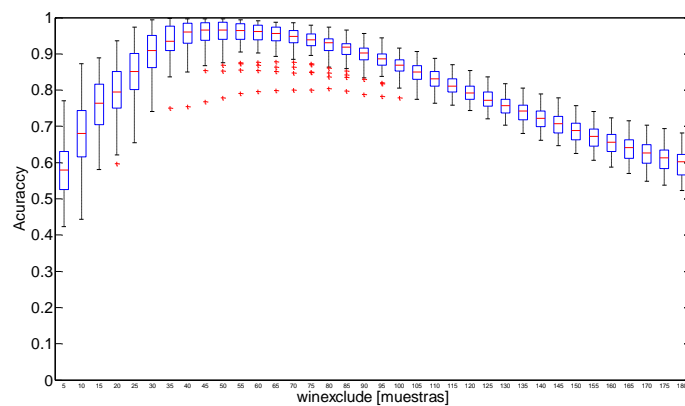
a)



b)



c)



d)

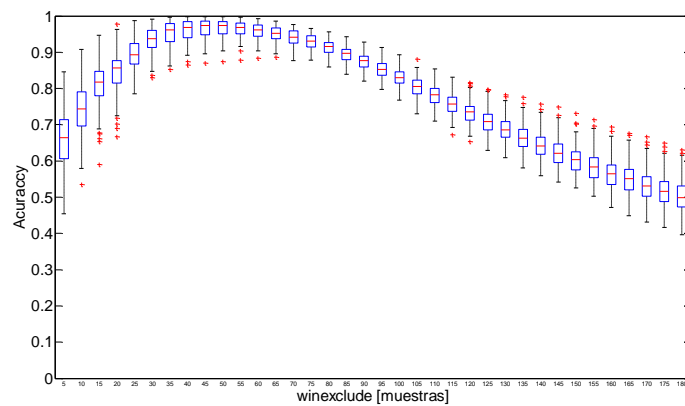
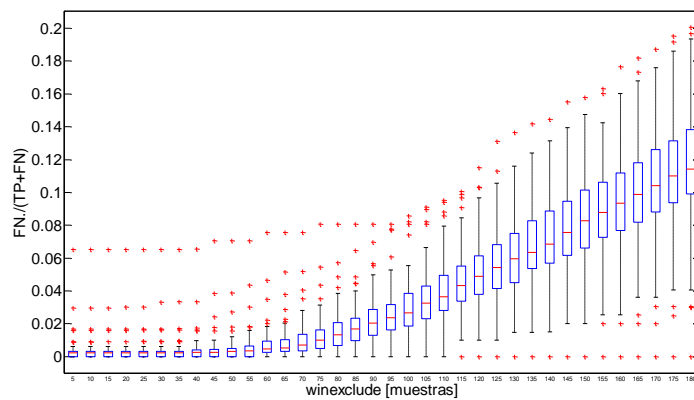
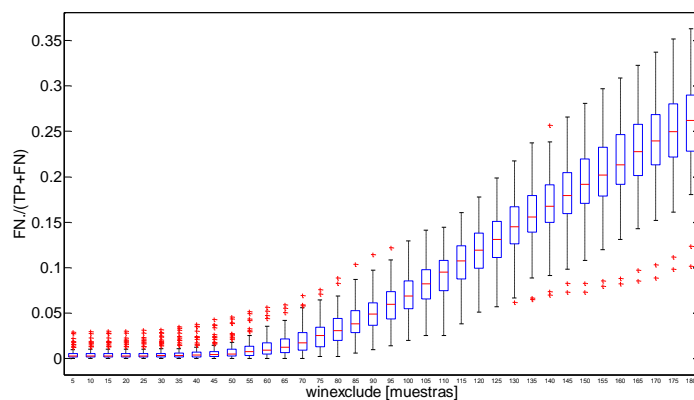


Figura 35. Accuracy variando el tamaño de la ventana de exclusión para distintas MVC.
a) MVC del 3% b) MVC del 6% c) MVC del 12% d) MVC del 24%.

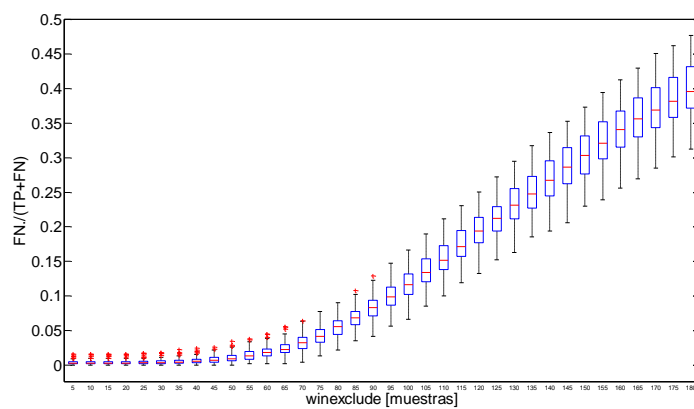
a)



b)



c)



d)

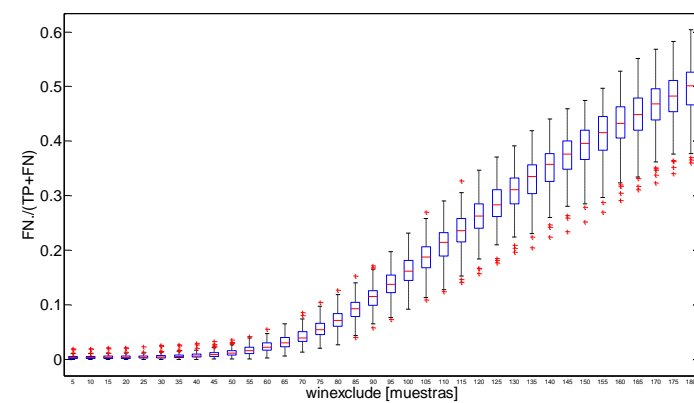
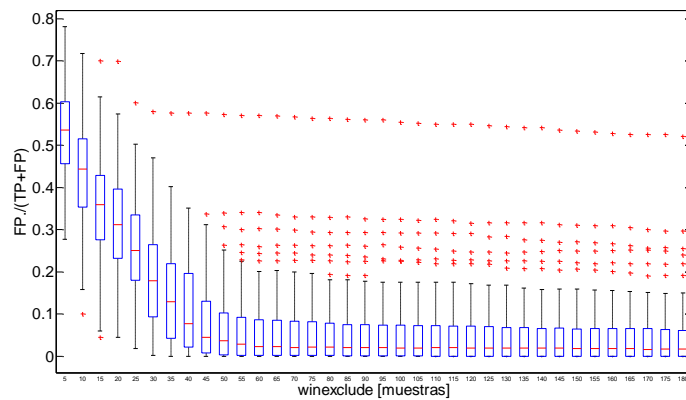
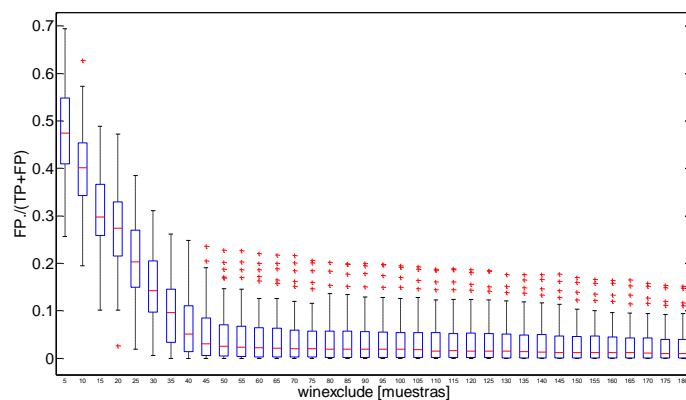


Figura 36. Porcentaje de falsos negativos variando el tamaño de la ventana de exclusión para distintas MVC. a) MVC del 3% b) MVC del 6% c) MVC del 12% d) MVC del 24%.

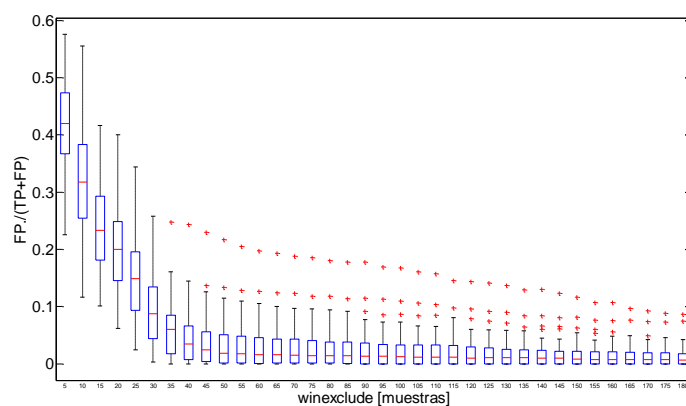
a)



b)



c)



d)

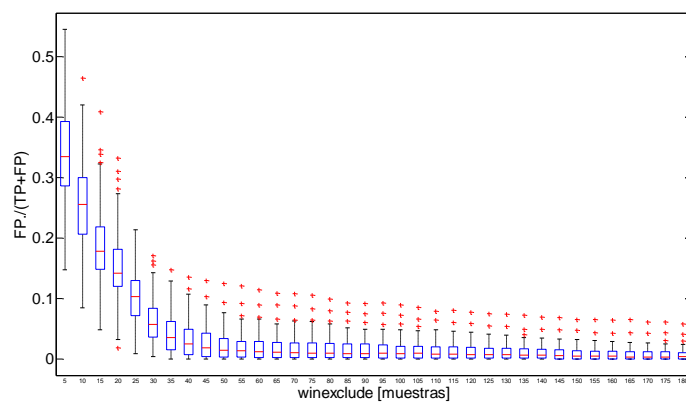


Figura 37. Porcentaje de falsos positivos variando el tamaño de la ventana de exclusión para distintas MVC. a) MVC del 3% b) MVC del 6% c) MVC del 12% d) MVC del 24%.

4.3.4 Variación de MVC

Una vez decidida la ventana óptima de exclusión analizaremos las figuras de mérito obtenidas con dicha ventana y variando el nivel de MVC.

Como vemos en la Fig. 39 (a), tenemos un *accuracy* constante por encima del 90% muy cercano al 100% incluso para niveles altos de MVC. Esto quiere decir que el número de FPs y FNs es muy bajo, por lo que el rendimiento de la segmentación es muy alto.

Falsos positivos

Si observamos la Fig. 39 (b), el porcentaje de FPs es muy bajo, por lo que no resulta perjudicial. Pero se aprecia un cambio grande según el MVC. Para niveles de contracción bajos el porcentaje de FPs está en torno al 10%-15% mientras que para niveles de contracción altos el porcentaje de FPs baja hasta un 5% aproximadamente. Esto puede ser debido al umbral de detección de Quiroga que hemos utilizado.

Cuando el nivel de contracción es bajo, nos encontramos con poca actividad en la señal por lo que el umbral se encuentra muy cerca de la línea base y esto significa que hay picos que superan dicho umbral pertenecientes a la señal de línea base y se detecte como posible MUP cuando en realidad se trata de ruido. En cambio si la MVC es mayor tendremos mayor actividad y por lo tanto un umbral mayor, siendo menos probable detectar un pico no perteneciente a la señal EMG.

En nuestro algoritmo, al generar el tren de referencia para la evaluación de la segmentación, hemos descartado ciertas MUs lejanas (MUPs pequeñas menores que el umbral de Quiroga) que posiblemente al sumarse o al añadir ruido sí superan el umbral. En la Fig. 38 se puede ver cómo los FPs detectados son picos muy cercanos al umbral, siendo muy probable que estos picos sean los potenciales de esas MUs lejanas que se han descartado al principio de la segmentación

Falsos negativos

Los falsos negativos suponen un menor número de disparos detectados, lo cual puede resultar perjudicial para el proceso de clasificación de los MUPs. Pero como vemos en la Fig. 39 (c), el porcentaje de FNs es muy bajo (por debajo del 2%) por lo que no va ser perjudicial.

Cuando aumentamos el MVC, los potenciales se encuentran cada vez más cerca, con lo que la ventana de exclusión no es lo suficientemente pequeña como para distinguir segmentos entre MUPs cercanos.

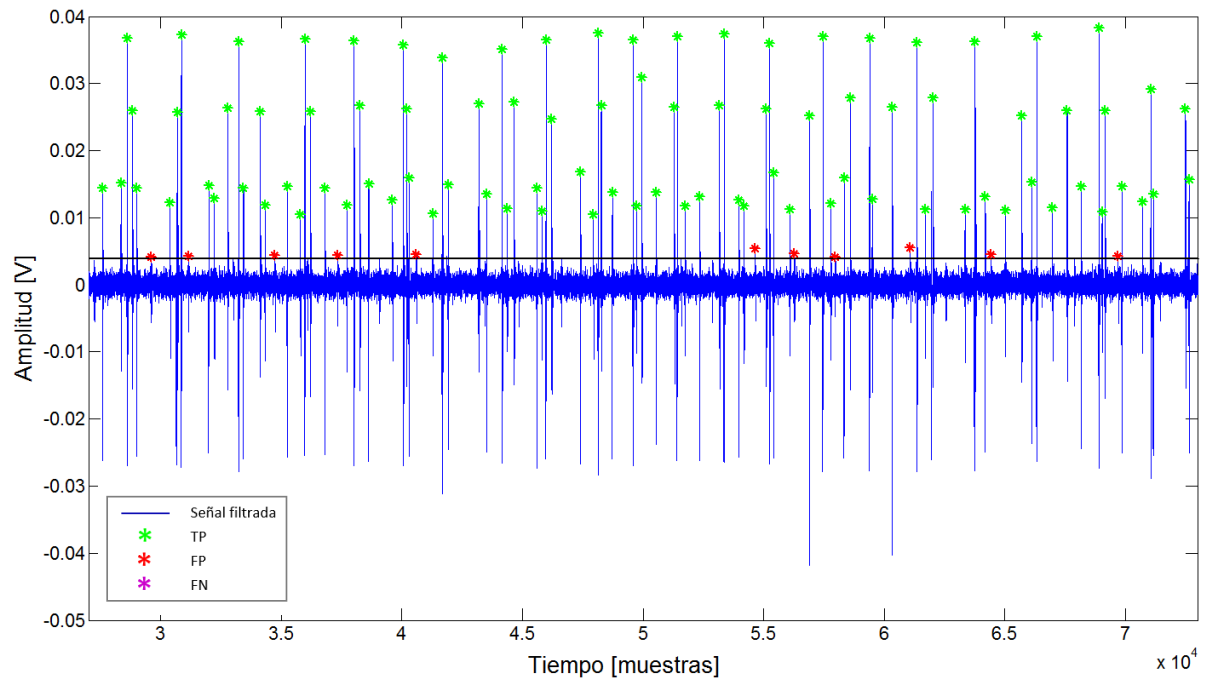


Figura 38. Ejemplo de casos donde ocurren FPs cercanos al umbral con un MVC del 3%

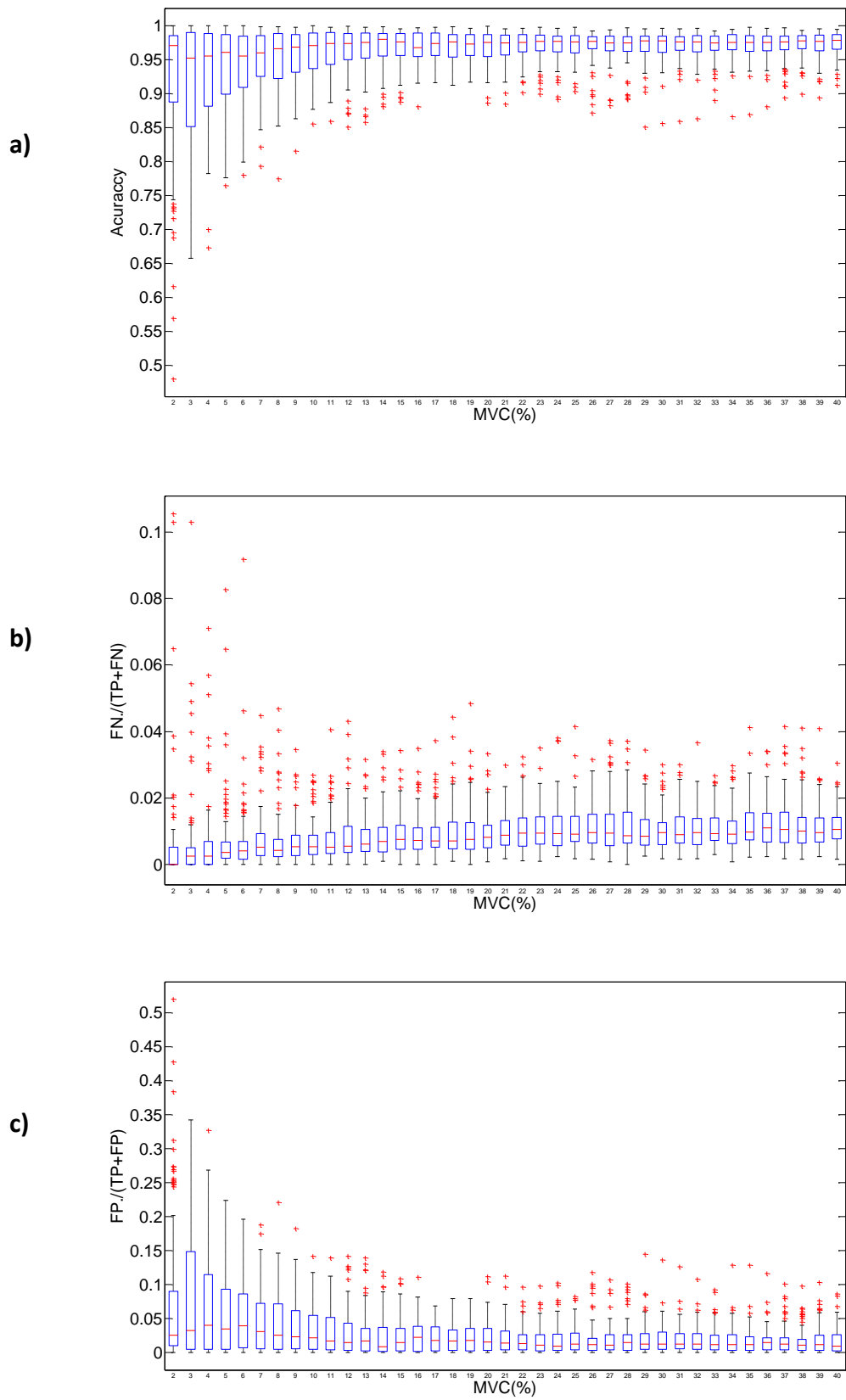


Figura 39. Variación del MVC con la ventana de exclusión de 45 muestras. a) Accuracy b) Porcentaje de FNs c) Porcentaje de FPs.

4.4 Clasificación

Tras finalizar el proceso de segmentación daremos comienzo al proceso de clasificación. Este proceso consiste en clasificar los MUPs que contienen los segmentos extraídos anteriormente.

4.4.1. Extracción de características

La extracción de características es el primer paso para llevar a cabo el proceso de clasificación. Esta etapa consiste en extraer la información más relevante, necesaria y suficiente para discriminar los MUP de una unidad motora respecto de otra. Al final de esta etapa, cada uno de los segmentos quedará representado por un vector de las características que se hayan escogido.

Para formar estos vectores existen diferentes técnicas de extracción de características, de las cuales las más destacadas son las basadas en análisis temporal (duración, tamaño, velocidad...) y las basadas en análisis frecuencial (potencia promedio, potencia máxima, análisis Cepstrum...). Es importante que las características seleccionadas contengan suficiente información y sean simples para facilitar el trabajo al algoritmo clasificador.

4.4.2 Clasificación no supervisada

En general, los problemas de clasificación se pueden atacar mediante dos técnicas: por un lado tenemos la clasificación supervisada donde a partir de un conjunto de observaciones ya clasificadas (conjunto de entrenamiento) se intenta clasificar un segundo conjunto de observaciones. Y por otro lado tenemos la clasificación no supervisada, en la que no se dispone de un conjunto previamente clasificado sino que a partir de las propiedades del conjunto de las observaciones se intenta clasificar según la similitud entre ellos.

La descomposición de EMG, donde no se conoce a priori la forma de onda de los MUPs a clasificar, entra en el rango de la clasificación no supervisada. Por lo tanto, a partir del vector de características obtenido previamente, el algoritmo de clasificación deberá ser capaz de determinar a qué MU pertenece cada segmento. Se trata de agrupar los segmentos en clases de tal forma que, los vectores de características de los segmentos de una misma clase tengan una alta similitud y a su vez, sean muy diferentes a las de otras clases. Cada uno de los MUP que contienen los segmentos quedará etiquetado por su correspondiente clase que representa a la MU que lo haya generado.

4.4.3 Obtención de prototipos y patrones de disparo

Una vez realizada la clasificación, las formas de onda de una misma clase se superponen y realizando un promedio de estas ondas se obtiene el prototipo que representa a la forma de onda del MUP de dicha clase. Y a partir del tiempo de adquisición de los segmentos clasificados se obtienen los patrones de disparo correspondientes a cada clase.

4.5 Evaluación de la clasificación

En este apartado se procede a evaluar única y exclusivamente el rendimiento del proceso de clasificación. La clasificación nos dará como resultado un vector de etiquetas que se evaluará en función de los aciertos que se hayan logrado respecto de las etiquetas reales.

El objetivo de la clasificación consiste en conocer a qué MU pertenecen los potenciales que contienen los segmentos. Para poder evaluar únicamente el proceso que realiza la clasificación debemos asegurarnos de que los segmentos que va a procesar el algoritmo para su posterior extracción de características y etiquetado, sean segmentos centrados en los instantes de tiempo reales que hemos seleccionado en el apartado 4.3.1, a los que llamaremos *Tref*. De esta forma, nos aseguramos de que la clasificación se realiza sobre todos los segmentos que el algoritmo de segmentación debería ser capaz de detectar.

Las etiquetas reales nos van a indicar a qué MU pertenecen realmente los potenciales que contienen cada uno de estos segmentos centrados en los instantes *Tref*. En nuestro entorno de simulación ya disponemos de esta información. Por lo tanto, como se muestra en la Fig. 40 se han etiquetado todos los disparos de cada *cell-array* de forma ordenada siendo dicha etiqueta la referente a la MU a la que pertenecen. Una vez hecho esto, todos los disparos se ordenan en el tiempo, y en ese mismo orden se obtiene el vector de las etiquetas reales que nos servirá de referencia para evaluar la clasificación.

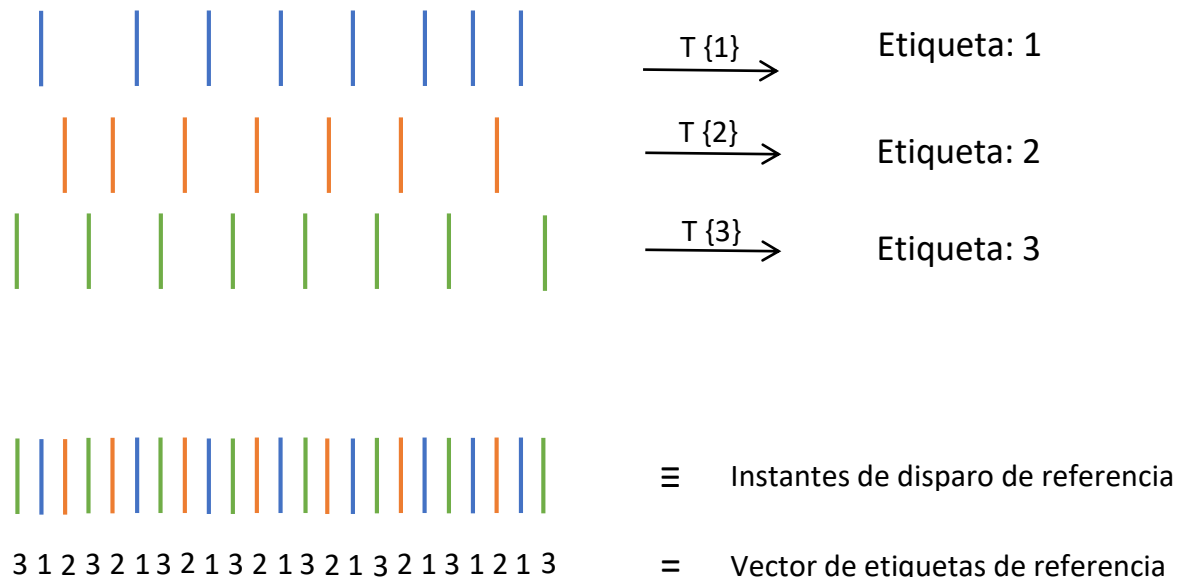


Figura 40. Obtención del vector de etiquetas de referencia.

Antes de dar comienzo a la evaluación del proceso de clasificación debemos realizar un último paso primordial para realizar la comparación de los vectores de etiquetas correctamente. Se trata de convertir las etiquetas que nos devuelve el algoritmo de clasificación a la misma numeración de las etiquetas que se tienen en el vector de referencia.

Es decir, en el proceso de extracción de características cuando el algoritmo encuentra un conjunto de MUPs con unas determinadas características en común y decide que pertenecen a la misma unidad motora, el algoritmo los etiqueta según el número de conjuntos que haya encontrado. Y al tratarse de una clasificación no supervisada, la etiqueta de la clase no tiene por qué coincidir con la del índice de la MU real. En la Fig. 41, se muestra un ejemplo donde el algoritmo ha detectado tres conjuntos de MUPs que cumplen con las mismas características y los etiqueta del 1 al 3 sin seguir ningún orden específico. Los instantes de tiempo etiquetados con un 1 (color naranja), son los mismos instantes de tiempo etiquetados con un 2 (etiqueta de referencia) en la Fig. 40. Por ello es necesario identificar los MUs obtenidos por el algoritmo de clasificación asociando los trenes de disparo.

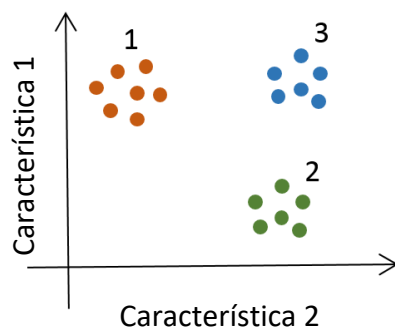


Figura 41. Etiquetado en el proceso de extracción de características.

Para saber a qué etiqueta debemos convertir cada una de ellas utilizaremos la siguiente función:

```
[I,A]=train_best_match_unique(T_,T)
```

Parámetros de entrada:

- $T_$: estructura *cell-array* que contiene los instantes de disparo ideales de referencia (en ms).
- T : estructura *cell-array* que contiene los instantes de disparo de la clasificación (en ms).

Parámetros de salida:

- I : Número de tren de $T_$ que se corresponde a cada tren del *cell-array* T .
- A : Valor de *accuracy* de cada tren en el *cell-array* T .

En la Fig. 42, se muestran 3 casos diferentes con los que nos podemos encontrar en el proceso de la extracción de características. La conversión de etiquetas finalmente se realizará en función del caso en el que nos encontremos mediante el parámetro \mathbb{I} :

- **Caso 1:** ocurre cuando el algoritmo de clasificación ha etiquetado un conjunto de MUPs que coincide con uno de los trenes de disparo de referencia, con lo cual dichos instantes pertenecen a una única MU. En este caso, simplemente convertiremos la etiqueta que nos devuelve el algoritmo de clasificación (C_{kclass}) en la que nos indica su correspondiente valor de \mathbb{I} .
- **Caso 2:** ocurre cuando el algoritmo de clasificación ha etiquetado en la misma MU un conjunto de MUPs que en realidad pertenecen a más de una MU diferente. Al pertenecer a más de una MU, tenemos en \mathbb{I} más de una etiqueta para convertir (una para cada MU). En estos casos, elegiremos el valor de \mathbb{I} que más *accuracy* nos ofrezca.
- **Caso 3:** ocurre cuando el algoritmo de clasificación etiqueta los MUPs en distintas MUs cuando en realidad todos ellos pertenecen a la misma MU. En este caso el valor de \mathbb{I} se repite para varios trenes de disparo, algunos teniendo un *accuracy* muy bajo. Nos quedaremos únicamente con el que mejor *accuracy* tenga, eliminando los demás trenes de disparo que tienen la misma etiqueta \mathbb{I} .

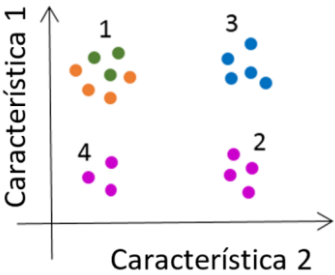
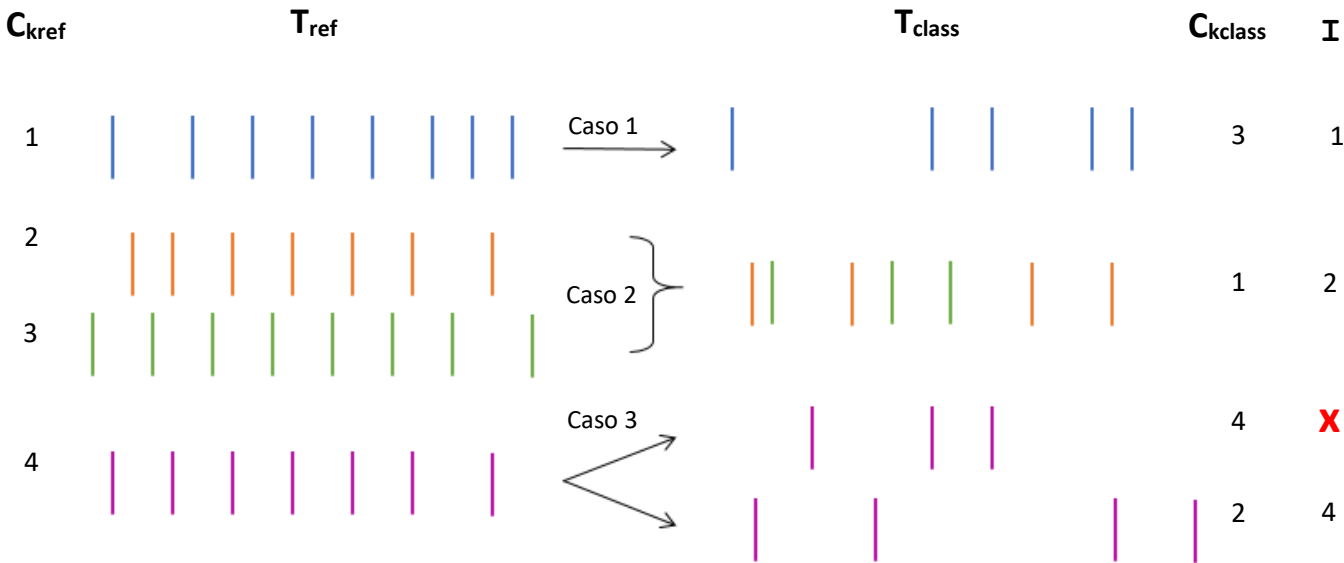


Figura 42. Ejemplo de los 3 casos que pueden ocurrir en el proceso de la extracción de características de la clasificación. En el lado izquierdo se muestran junto a sus respectivas etiquetas (C_{kref}) los trenes de disparo reales (T_{ref}), en formato cell-array, a partir de los cuales se extraen los segmentos y en el lado derecho se observan los trenes de disparo (T_{class}), en formato cell-array, correspondientes a los segmentos que el algoritmo ha podido etiquetar (C_{kclass}), además de la etiqueta a la que se va a convertir cada uno (I).

Una vez hecha la conversión podremos comparar los dos vectores de etiquetas disparo a disparo y determinar la calidad del sistema obteniendo los aciertos y errores. La calidad del sistema viene dado por el valor de *accuracy* (A), el cual se define como:

$$A = \frac{\text{nº de aciertos}}{\text{nº disparos total}} \quad (2)$$

En la Fig. 43, se muestra un ejemplo del proceso de comparación de los vectores de etiquetas. Los disparos que el algoritmo de clasificación no ha podido etiquetar se indican con un cero. En el ejemplo se puede observar que los disparos correspondientes a las MUs 2 y 3, el algoritmo los ha clasificado en una única MU (caso 2) con la etiqueta 2 en este caso, por ser el que mayor *accuracy* nos ofrecía. Por lo que prácticamente la mitad de los disparos de esta MU se detectarán erróneas.

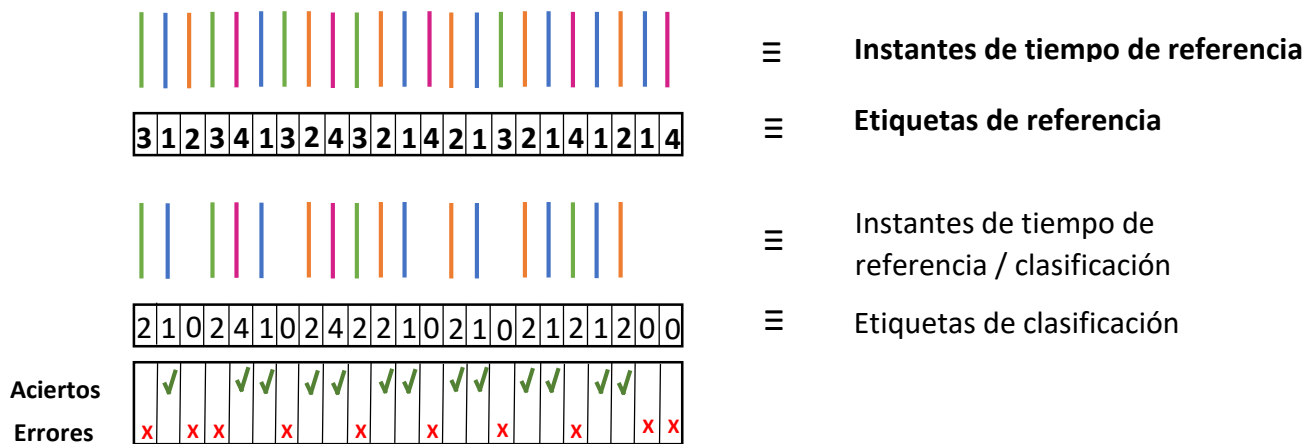


Figura 43. Diagrama de definición del valor de *accuracy* en el caso de la clasificación.

4.5.1 Descripción del experimento

Para cuantificar el rendimiento del algoritmo de clasificación, se realiza un experimento donde se generan 100 músculos de distinto tamaño. Para cada realización se obtiene:

- Señal EMG registrada por el electrodo de aguja
- Trenes de disparo de referencia junto a las etiquetas de referencia
- Segmentos extraídos a partir de los trenes de referencia
- Vector de características basada en el análisis temporal, concretamente las siguientes características (elección según el criterio de [7]): máxima pendiente, amplitud y área.
- Etiquetas obtenidas por el algoritmo de clasificación
- Etiquetas \mathbb{I}
- Vector de aciertos (*accuracy*).

Estas realizaciones se calcularán variando un único parámetro: el MVC. En este caso, se hará una variación del 2 al 20% con incrementos de un 1%.

El objetivo de este experimento es evaluar el rendimiento de diferentes algoritmos de clasificación y compararlos para decidir cuál de ellos es mejor. En este proyecto se han evaluado tres aproximaciones diferentes de clasificación no supervisada: k-Means, jerárquico y Kohonen. Se realizará este experimento para cada uno de ellos.

4.5.2 Influencia de las superposiciones

Como ya se ha comentado anteriormente, el incremento del nivel de MVC implica una mayor probabilidad de que dos unidades motoras disparen muy próximas o al mismo tiempo, lo cual supone una superposición de sus correspondientes MUPs. En estos casos, el algoritmo de segmentación extraerá segmentos con formas de onda difíciles de vincular a otros potenciales dando lugar a una mala clasificación.

En este proyecto no se ha implementado ningún método que localice y resuelva las superposiciones que contienen los segmentos, por lo que el funcionamiento de los algoritmos de clasificación que veremos en este capítulo, se evaluarán de acuerdo a las limitaciones que estas superposiciones suponen. Para ello, se ha estimado una curva del porcentaje de MUPs respecto del nivel de MVC que no se ven involucrados en superposiciones o que se ven involucrados sin que ello conlleve una mala clasificación (Fig. 44). Esto quiere decir que para considerar un buen rendimiento del algoritmo de clasificación, éste deberá ser capaz de clasificar correctamente al menos un porcentaje de MUPs muy próximo a esta curva. De esta forma, podremos ser capaces de evaluar los algoritmos de forma más objetiva. Como es lógico se obtiene una curva que decrece según aumenta el MVC.

Esta curva se ha aproximado teniendo en cuenta las unidades motoras activas que solapan la posición de los MUPs que ha capturado el electrodo en una posición determinada.

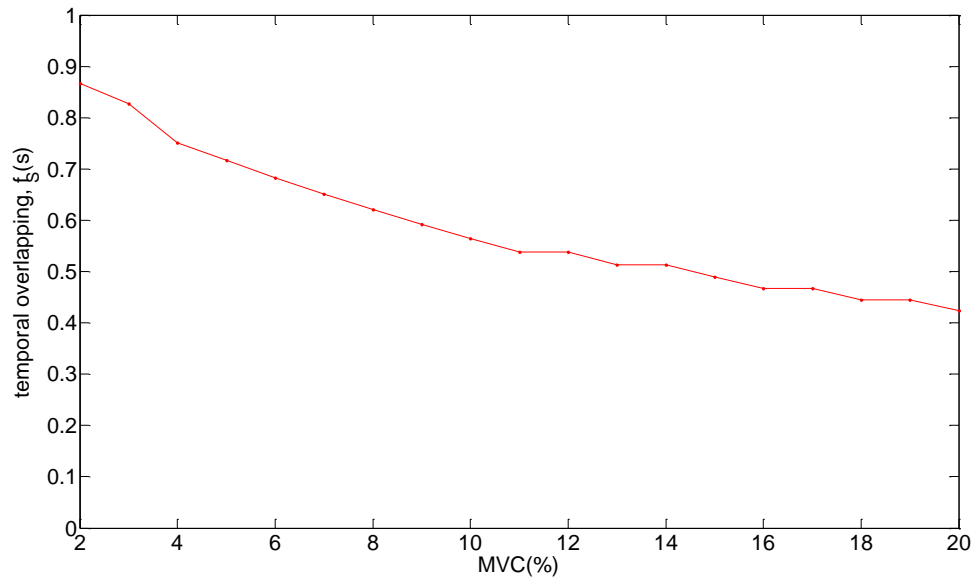


Figura 44. Curva aproximada del porcentaje de MUPs sin superposiciones respecto del MVC.

4.5.3 Algoritmos de clasificación

En este apartado, se describe muy brevemente el funcionamiento de los tres algoritmos de clasificación que se van a evaluar posteriormente.

4.5.3.1 k-Means

k-Means es un algoritmo particional que divide un conjunto de datos en k particiones, donde cada partición representa una clase. Para ello, el procedimiento que se sigue consta de tres fases: inicialización, asignación y actualización.

- *Inicialización*: inicialmente se debe de estimar el número de clases (k) en el espacio de características, asignando a cada un centroide que representa a los centros de cada partición.
- *Asignación*: a cada segmento se le asignará el centroide más próximo y se volverán a calcular los centros de las k clases.
- *Actualización*: se repetirá el punto anterior hasta que se haya llegado al número máximo de iteraciones.

El resultado de este tipo de clasificación depende en gran medida de la estimación del número de clases que se hace al inicio de su ejecución.

4.5.3.2 Kohonen

El algoritmo de Kohonen realiza la clasificación utilizando una red neuronal de un determinado tamaño que dividirá el espacio de características en diferentes grupos. Después, el usuario definirá el número mínimo de segmentos que deben contener los

grupos. Todos aquellos grupos que tengan menor número de segmentos que ese mínimo serán eliminados. En nuestro caso, se ha definido una red neuronal de 3x3 y el número mínimo de segmentos de cada grupo es de 10. Esto implica que la red neuronal intenta particionar el espacio de características en 9 clases de las cuales pueden acabar eliminándose aquellas con menos de 10 segmentos asignados.

4.5.3.3 Jerárquico

En el algoritmo de clasificación jerárquica, los objetos (segmentos) del espacio de características se organizan mediante una jerarquía con estructura de árbol. Se trata de ir agrupando en cada paso aquellos dos objetos más cercanos formando una estructura conocida como dendrograma (Fig. 45). La configuración de la jerarquía termina cuando se forma una única clase que engloba todas las observaciones del espacio de características. Por último, se traza una línea a partir del percentil 95 de la distancia y quedan definidas las clases a las que pertenecen los segmentos.

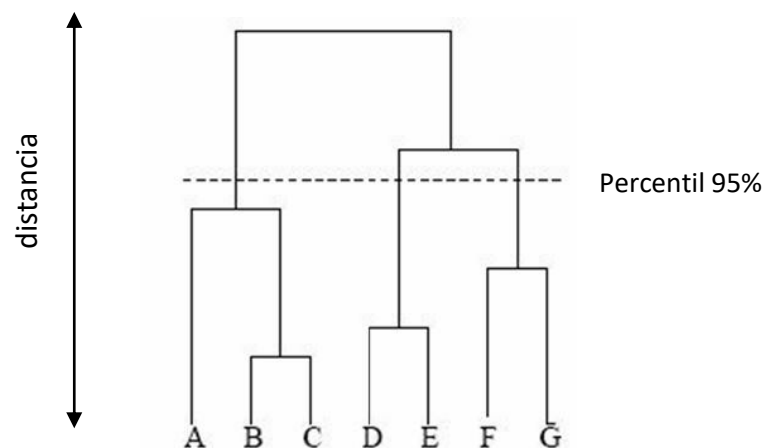


Figura 45. Representación de un dendrograma donde A, B Y C pertenecen a la clase 1, D y E a la clase 2 y F y G a la clase 3.

4.5.4 Comparación de los diferentes algoritmos de clasificación

A continuación compararemos los algoritmos descritos anteriormente y analizaremos cuál de ellos tiene mejor rendimiento.

En la Fig. 46 se muestran los resultados del *accuracy* frente al MVC a modo de box plot para cada algoritmo. Sobre cada uno de estos resultados se ajusta la curva que aproxima el porcentaje de MUPs no superpuestos que nos servirá de guía para determinar y comparar el rendimiento de cada uno de ellos.

En general, el porcentaje de aciertos de los tres algoritmos se aproxima mucho a esta curva, por lo que se podría decir que se consigue una buena estimación en todos ellos. Sin embargo, si observamos los resultados más detenidamente vemos que a bajos niveles de contracción la estimación no es tan buena en los algoritmos de k-Means (Fig. 46 (a)) y Kohonen (Fig. 46 (b)).

En el caso del k-Means, uno de sus principales inconvenientes es la necesidad de establecer el número de clases que se estiman al inicio de su ejecución. Este hecho perjudica al rendimiento del algoritmo ya que el número de clases que hay en la señal no se conoce a priori. Si se estima un número de clases superior al que en realidad se tiene, al tener pocos potenciales es más probable que la clasificación resulte errónea que en el caso de tener muchos potenciales. Posiblemente este inconveniente sea la razón por la que a bajos niveles de contracción, la exactitud sea tan dispersa.

En cuanto al algoritmo de Kohonen, realiza la mejor estimación a altos niveles de contracción pero la peor a bajos niveles. Esto probablemente sea debido al parámetro que se define como el mínimo número de segmentos que debe contener cada grupo de la red neuronal que en nuestro caso ha sido 10. A bajos niveles de contracción, los potenciales de una misma MU pueden clasificarse en más de un grupo de la red neuronal que seguramente contengan menos de 10 potenciales y si esto es así, el algoritmo los descartará directamente. Esto implica un menor porcentaje de acierto como se refleja en la Fig. 46 (b).

Por último, el algoritmo de clasificación jerárquica presenta una buena estimación tanto para bajos niveles de MVC como para altos. Siendo la curva del porcentaje de MUPs no superpuestos prácticamente la misma que la curva de medianas de su box plot.

Para comparar de forma más precisa el rendimiento de estos algoritmos, se ha realizado una representación de las curvas del primer y tercer percentil (25% y 75%) de las distribuciones (Fig. 47 (a)), además de la curva de las medianas. Las curvas obtenidas confirman lo dicho, siendo Kohonen el algoritmo que mejores valores de *accuracy* obtiene a altas MVCs y los peores a bajas MVCs. Los algoritmos de k-Means y el jerárquico presentan resultados muy similares entre sí con la única diferencia de que k-Means presenta un menor rendimiento a bajos niveles de contracción. Por lo tanto, claramente el algoritmo jerárquico es con el que mejores resultados vamos a obtener siendo el único de los tres que proporciona un buen rendimiento a lo largo de todo el rango de niveles de contracción.

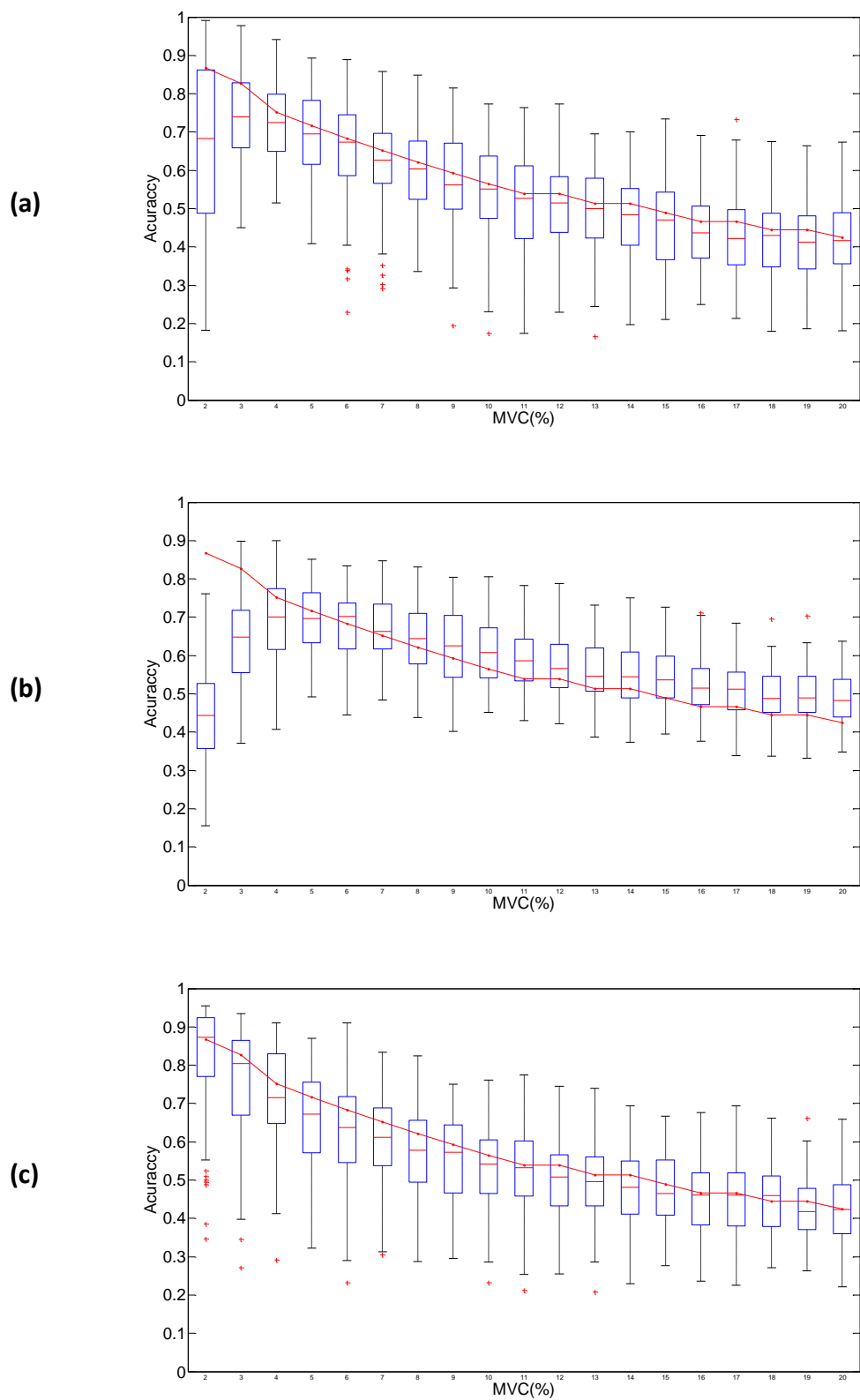
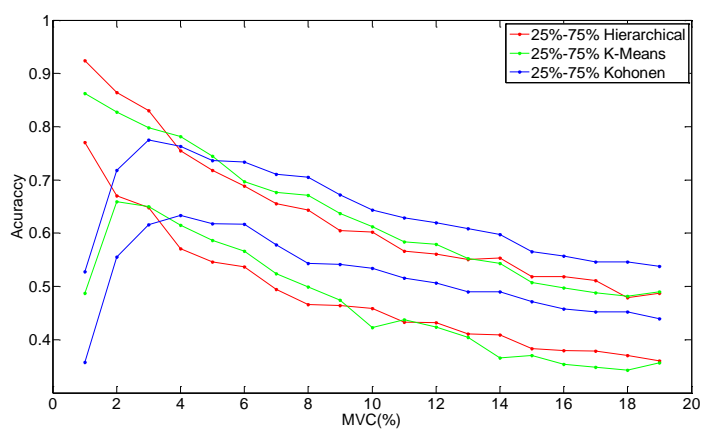
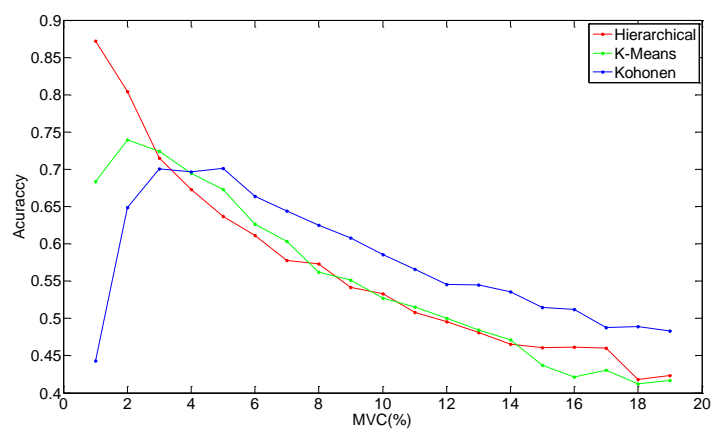


Figura 46. Diagramas de caja de accuracy de los diferentes algoritmos de clasificación.
a) K-Means b) Kohonen c) Jerárquico



(a)



(b)

Figura 47. Comparación de los tres diferentes algoritmos. a) Accuracy en el rango de los percentiles del 25 al 75% b) Accuracy según la mediana.

4.6 Evaluación del proceso completo: segmentación y clasificación

En este apartado se procede a evaluar de forma conjunta el rendimiento de los procesos de segmentación y clasificación. En general, el método descrito aquí permite evaluar, como haremos con el algoritmo de Montreal, métodos de descomposición automática completos, independientemente de los procesos que incluyan (como resolución de superposiciones, por ejemplo). Para determinar la calidad del resultado, se tomarán como referencia los trenes de disparo en formato *cell-array* con sus respectivas etiquetas como hemos visto en el apartado anterior.

En este caso, para poder evaluar el proceso completo partimos de los instantes de tiempo obtenidos por el algoritmo de segmentación para extraer los segmentos. El algoritmo de clasificación se encargará de extraer las características de dichos segmentos y determinar la MU a la que pertenecen mediante un vector de etiquetas.

Los disparos que han sido etiquetados con el mismo número de etiqueta se van a agrupar en un tren de disparos en formato *cell-array* para compararlos con el tren de disparos de referencia que le corresponde. Como hemos visto en el apartado anterior las etiquetas que se obtienen en la clasificación hace falta traducirlos al mismo tipo que los de referencia. La agrupación de los trenes de disparo que acabamos de comentar se realizará según la etiqueta ya convertida. Cada tren de disparos se compara con su tren de referencia mediante los parámetros TP, FP y FN calculados como se muestra en la Fig. 48.

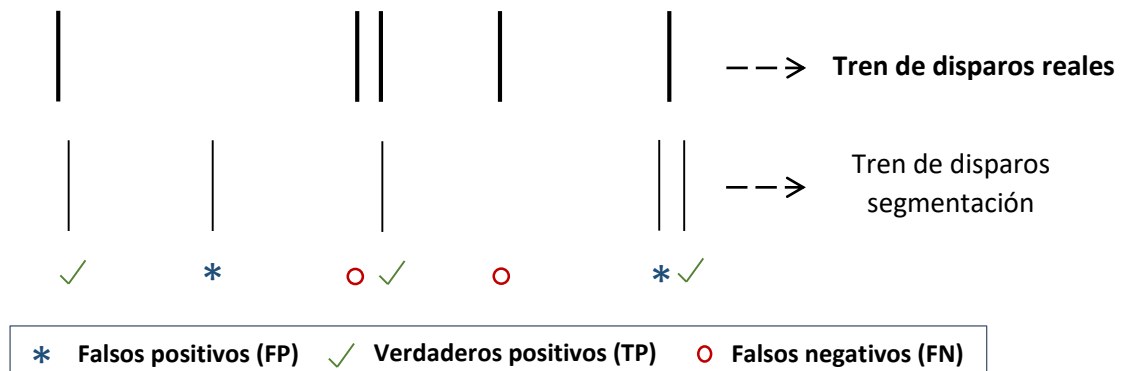


Figura 48. Diagrama de definición del comportamiento de los FP, FN y TP en el caso de la clasificación.

Una vez calculados estos parámetros se define el valor de *accuracy* de la siguiente manera:

$$A_k = \frac{TP}{TP+FN+FP} ; \quad \text{siendo } k = n^{\circ} \text{ de la UM} \quad (3)$$

Una vez obtenido el valor de *accuracy* de cada tren de disparos, es decir para cada MU, se calculará la media de todas ellas para finalmente obtener el valor de *accuracy* final que determine la calidad del sistema completo:

$$A_{media} = \frac{1}{K} \sum A_k; \quad \text{siendo } k \text{ el n}^\circ \text{ de UMs total} \quad (4)$$

4.6.1 Descripción del experimento

Para cuantificar el rendimiento del proceso completo, se realiza un experimento donde se generan 100 músculos de distinto tamaño. Para cada realización se obtiene:

- Señal EMG registrada por el electrodo de aguja
- Trenes de disparo de referencia junto a las etiquetas de referencia
- Instantes de disparo obtenidos por el algoritmo de segmentación.
- Segmentos extraídos a partir de los instantes anteriormente obtenidos
- Vector de características basada en el análisis temporal, concretamente las siguientes características (elección según el criterio de [7]): máxima pendiente, amplitud y área.
- Etiquetas obtenidas por el algoritmo de clasificación
- Figuras de mérito: *Accuracy*, FPs, FNs.

Estas realizaciones se calcularán variando un único parámetro: el MVC. En este caso, se hará una variación del 2 al 20% con incrementos de un 1%.

El objetivo de este experimento es evaluar el rendimiento del proceso completo de descomposición que consiste en: el algoritmo de segmentación implementado en este proyecto (apartado 4.2) junto a los diferentes algoritmos de clasificación no supervisada (k-Means, Kohonen y jerárquico). Además se evaluará también a modo de referencia la evaluación del sistema de descomposición completo de Montreal.

4.6.2 Influencia de los errores de la segmentación

La segmentación es un proceso crítico en el sistema completo de descomposición, dado que si no se extraen los segmentos correctamente difícilmente se podrán clasificar. Por ello, es preciso conseguir el menor número de FNs y FPs posible.

Como ya se comentó en el apartado 4.3.4, los FPs generados en la segmentación podrían ser MUPs lejanos que nuestro algoritmo detecta al sumarse con el ruido. Si esto es así lo más probable es que el algoritmo de clasificación encuentre cierta similitud entre ese conjunto de MUPs lejanos y los agrupe en una misma clase. Probablemente la clasificación sea correcta pero en nuestro método de evaluación las etiquetas de referencia no incluyen las correspondientes a dichos MUP, de modo que se evalúan

todos ellos como FPs. Puede suceder también que los FPs de la segmentación se mezclen junto a clases a las que no pertenecen provocando en el resultado final otro FP.

Los FNs de la segmentación suponen al algoritmo de clasificación un menor número de segmentos a clasificar.

4.6.3 Influencia de las superposiciones

Como ya se ha comentado en el apartado anterior, un segmento que contiene una superposición de MUPs el algoritmo de clasificación lo intentará clasificar como si fuera un único MUP ya que no resuelve las superposiciones. En estos casos pueden ocurrir 2 cosas:

- El algoritmo de clasificación no encuentra similitud entre la forma de onda de la superposición y los demás MUPs. Por lo que se etiqueta con un 0 y se genera un falso negativo por cada MU a la que pertenece dicha superposición.
- El algoritmo de clasificación encuentra cierta similitud entre la forma de onda de la superposición y alguna otra MUP que coincide con alguno de los MUP que forman la superposición generando un TP en ese instante y un FN en los instantes de los demás MUP que forman dicha superposición. Sin embargo en el caso de que la similitud la encuentre con un MUP que no coincida con ninguno de la superposición se generará un FP.

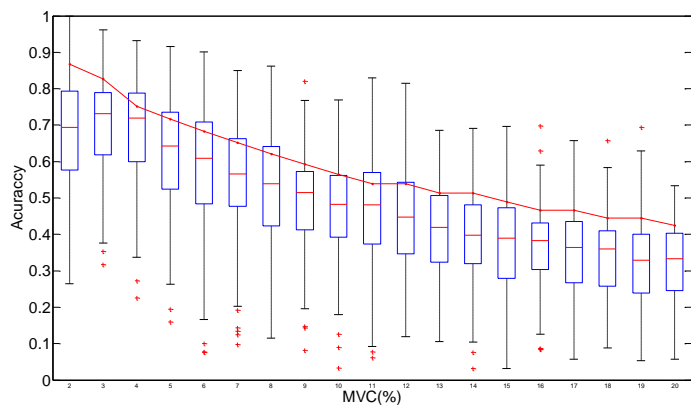
4.6.4 Comparación de los distintos algoritmos

En este apartado, se comparan los tres sistemas de descomposición completas donde el algoritmo de clasificación es lo único que les diferencia. Y estos a su vez se van a comparar con el sistema de descomposición de Montreal.

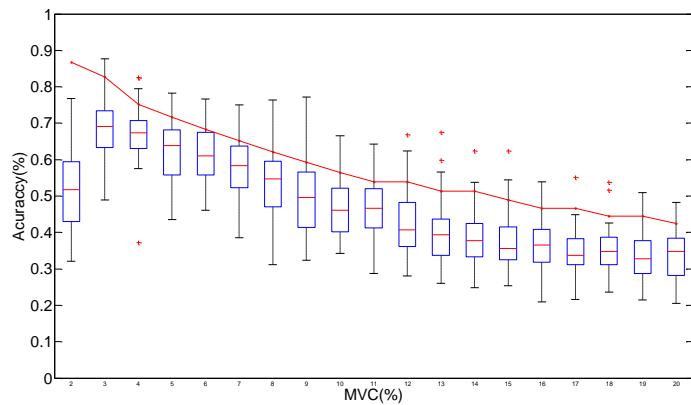
En la Fig. 49 y 50, se muestran los resultados de las figuras de mérito obtenidas para cada caso a modo de box plot junto a la curva aproximada de los MUP no superpuestos explicada en el apartado 4.5.2. Como se puede ver, estos resultados son muy similares a los de la Fig. 46 salvo que ahora la influencia del algoritmo de segmentación refleja una ligera bajada en los valores de *accuracy*. El algoritmo que más ha sufrido esta bajada es el de Kohonen, donde la curva aproximada de los MUP no superpuestos se encuentra al borde de la curva del percentil del 75%. El algoritmo jerárquico en cambio, es el menos influenciado por la segmentación.

Hay que tener en cuenta que como vimos en el apartado 4.3.4, el algoritmo de segmentación se evalúa con un alto rendimiento con valores de *accuracy* superiores al 90% en todo el rango de niveles de contracción. Por lo que la influencia que pueda tener respecto a la clasificación es mínima. Y conociendo que el algoritmo de clasificación más óptimo es el jerárquico, era de esperar que el sistema de descomposición de mayor rendimiento también sería el jerárquico. Respecto al porcentaje de FNs y FPs detectados, lógicamente incrementan a mayor valor de MVC debido a una mayor probabilidad de superposición. Y en los 3 casos se presentan curvas similares.

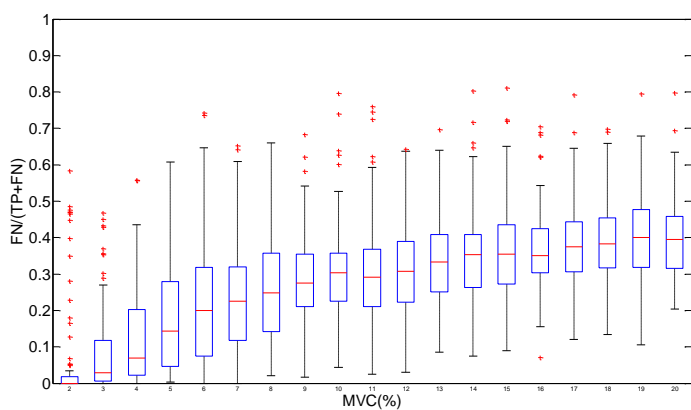
Si comparamos estos tres sistemas con el sistema de descomposición de Montreal, vemos sobre todo la gran influencia que suponen las superposiciones teniendo en Montreal un porcentaje de accuracy bastante superior a los demás. En la Fig. 51 se muestra una comparación de los tres sistemas junto al de Montreal, empleando las curvas de los percentiles del 25% y el 75% (Fig.51 (a)) y las curvas de la mediana del box plot (Fig. 51 (b)) donde se aprecia la gran diferencia que hay entre un sistema que no resuelve superposiciones frente a una que sí lo hace.



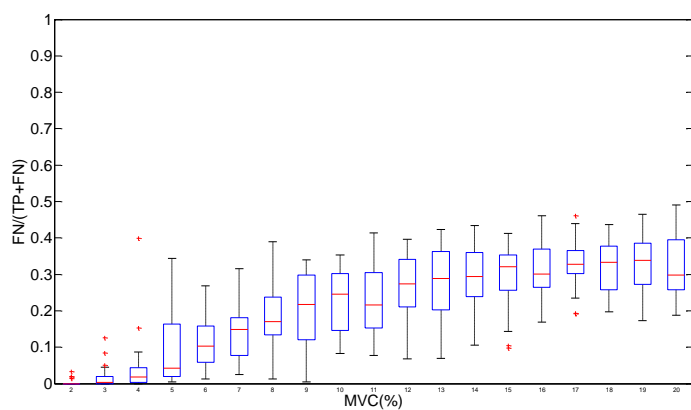
(a)



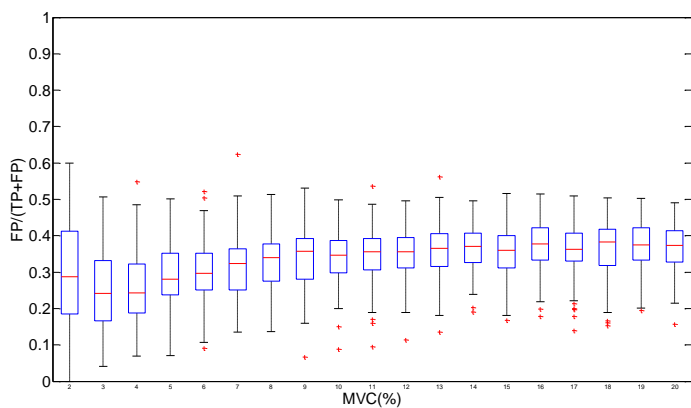
(b)



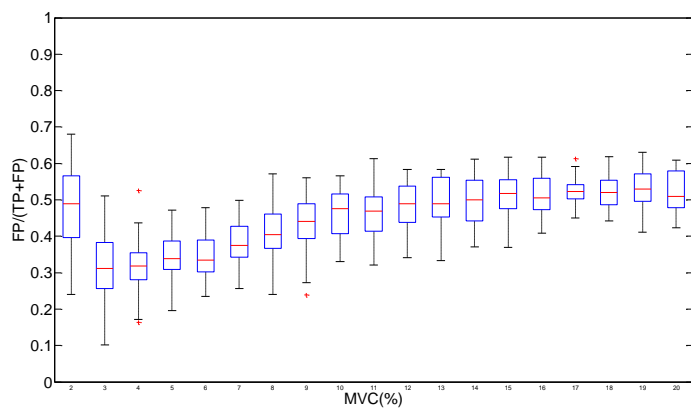
(c)



(d)

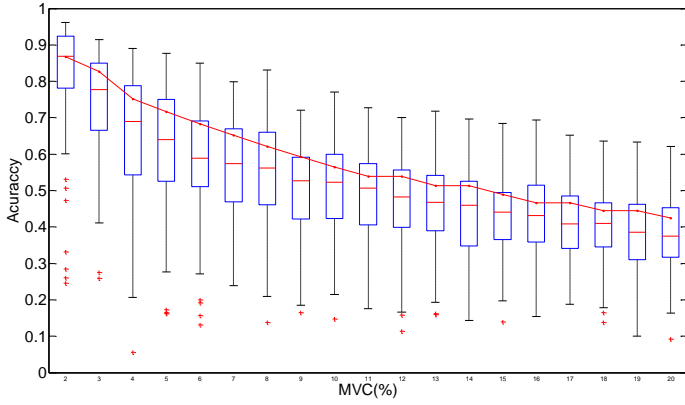


(e)

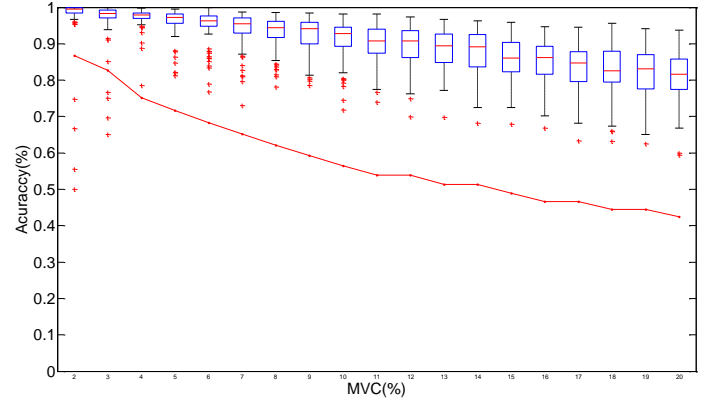


(f)

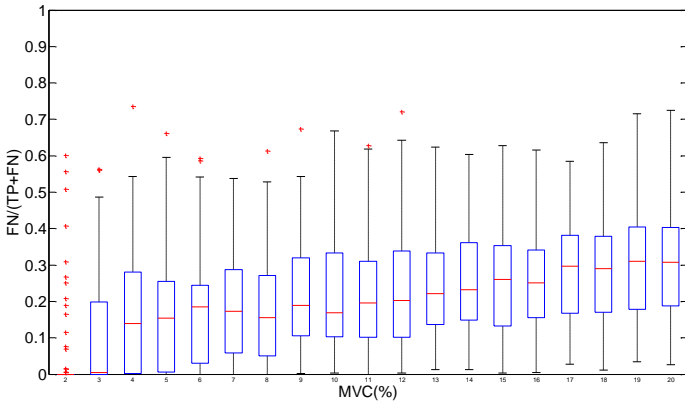
Figura 49. Diagramas de cajas de las figuras de mérito del proceso completo: proceso de segmentación implementado en este proyecto junto al algoritmo de k-Means en la columna izquierda y el algoritmo de Kohonen a la derecha. a) Accuracy K-Means b) Accuracy Kohonen c) FNs K-Means d) FNs Kohonen e) FPs K-Means f) FPs Kohonen



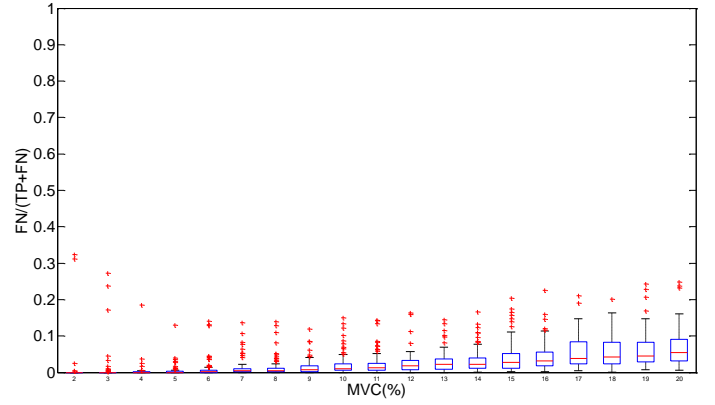
(a)



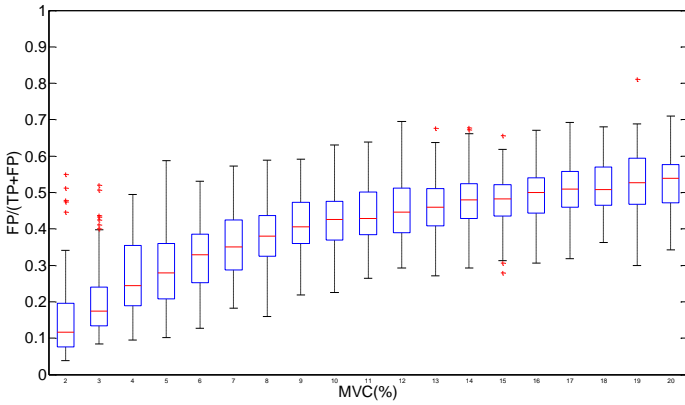
(b)



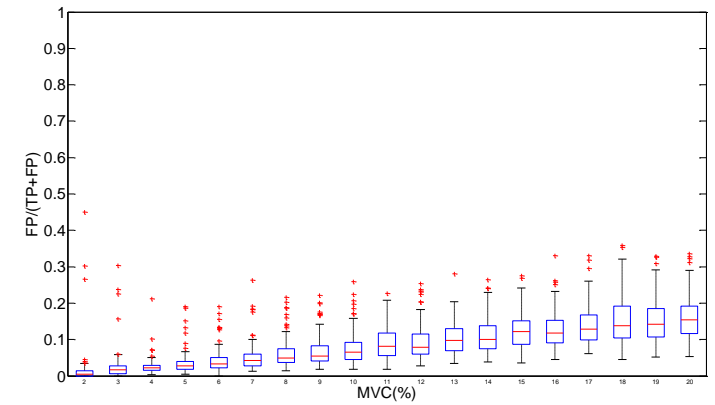
(c)



(d)



(e)



(f)

Figura 50. Diagramas de cajas de las figuras de mérito del proceso completo de descomposición: proceso de segmentación implementado en este proyecto junto a al algoritmo jerárquico en la columna izquierda. En la columna izquierda descomposición de Montreal a) Accuracy jerárquico b) Accuracy Montreal c) FNs jerárquico d) FNs Montreal e) FPs jerárquico f) FPs Montreal.

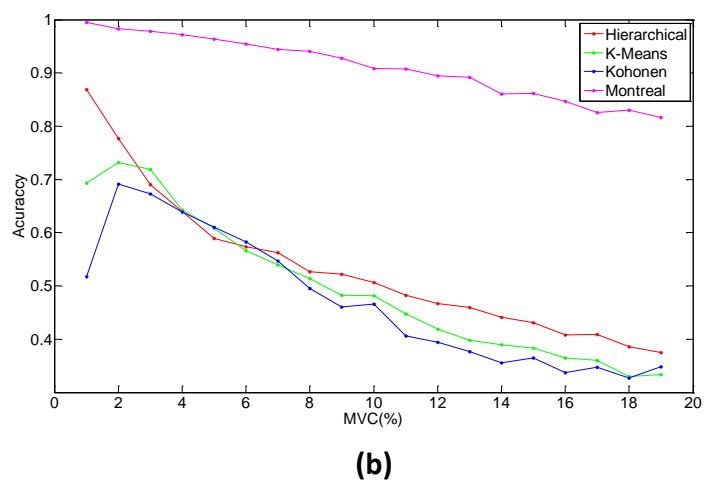
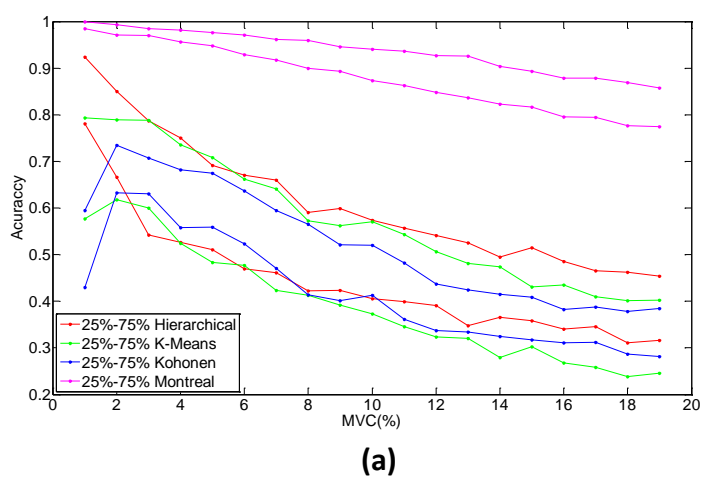


Figura 51. Comparación de los tres diferentes sistemas, siendo el algoritmo de segmentación el implementado en este proyecto y los algoritmos de clasificación: K-Means, Kohonen y jerárquico. Además del sistema de descomposición completo de Montreal. a) Accuracy en el rango de los percentiles del 25 al 75% b) Accuracy según la mediana.

Capítulo 5. Conclusiones y líneas futuras

5.1 Conclusiones

El objetivo principal de este proyecto ha sido diseñar e implementar métodos para evaluar la calidad de un sistema de descomposición automática EMG. Un sistema de descomposición, está formado por dos etapas importantes que influyen en el rendimiento del sistema completo: segmentación y clasificación. Por ello, además de implementar un método que evalúa el sistema completo, se ha implementado también para cada etapa un método de evaluación de acuerdo a las condiciones que requiere cada una.

En el proceso de la segmentación, el objetivo es desarrollar un método que evalúe la capacidad que tiene un algoritmo para detectar los segmentos que contienen los MUP de la señal EMG. Estos segmentos se han evaluado según los instantes de disparo sobre los cuales están centrados, comparándolos con los instantes de disparo de referencia que representan a todos los MUP de la señal que el algoritmo debería ser capaz de detectar. Este algoritmo no es capaz de diferenciar MUPs superpuestos, entendiendo la superposición como un único MUP que posteriormente se deberá resolver, de forma que el método de evaluación de este proceso requiere que no se penalice la superposición de MUPs.

Mediante el método desarrollado, se ha tenido la oportunidad de evaluar la calidad de uno de los algoritmos de segmentación disponibles en el *toolbox* y según los resultados obtenidos se ha realizado una propuesta de una versión mejorada. El método nos ha servido además para verificar que se ha logrado dicha mejora.

En cuanto al proceso de clasificación, el objetivo es implementar un método capaz de evaluar el rendimiento de los algoritmos para etiquetar los MUPs que contienen los segmentos. La evaluación se ha realizado comparando una a una las etiquetas obtenidas por el algoritmo con las etiquetas de referencia. Antes de llegar al proceso de clasificar los segmentos, es importante que se hayan resuelto las superposiciones de MUPs que pudieran contener algunos de los segmentos. En este proyecto no se ha implementado ningún método que las resuelva, por lo que la evaluación de los algoritmos de clasificación se ha realizado teniendo en cuenta la limitación que suponen estas superposiciones.

En la evaluación del sistema de descomposición completo, el proceso de segmentación es decisivo para obtener un buen resultado ya que si no se extraen los segmentos correctamente, difícilmente se podrán clasificar. Además, se ha podido comprobar la gran influencia que suponen las superposiciones en el resultado final ya que si no han sido resueltas, el algoritmo de clasificación no es capaz de encontrar similitudes con otros MUPs y esto implica errores en la clasificación.

5.2 Líneas futuras

Para próximos proyectos, se plantean algunas cuestiones de interés que no han sido resueltas:

- Analizar la influencia de la selección de las características del MUP en el dominio temporal en el resultado del proceso de clasificación para determinar cuáles son las más discriminativas.
- Analizar la influencia de los parámetros libres de los métodos de clasificación (número de clases, umbral de distancia, umbral de purga de clases, etc.) en el rendimiento de los mismos.
- Implementar un método que resuelva las superposiciones y estudiar la ganancia que suponen en el resultado final. Tal y como se ha visto por la comparación con el algoritmo de Montreal, se espera que tenga un impacto grande en la mejora de resultados.

Capítulo 6. Bibliografía

- [1] Rodríguez-Falces J. (2013) A novel approach to teach the generation of bioelectrical potentials from a descriptive and quantitative perspective.
- [2] A.J. Fuglevand, D.A. Winter, A.E. Patla, Models of recruitment and rate coding organization in motor-unit pools, J. Neurophysiol. 70 (1993) 2470.
- [3] De Luca C.J., Hostage E.C.: Relationship between firing rate and recruitment threshold of motoneurons in voluntary isometric contractions, J Neurophysiol. 2010 Aug; 104(2): 1034- 46.
- [4] Navallas J, Malanda A, Gila L, Rodríguez J, Rodríguez I. (2009) Mathematical analysis of a muscle architecture model
- [5] Navallas J, Malanda A, Gila L, Rodríguez J, Rodríguez I (2009) Comparative evaluation of motor unit architecture models
- [6] Navallas J, Stålberg E, Rodríguez J. (2012). Scanning Electromyography, "EMG methods for evaluation muscle and nerve function" Mr. Mark Schwartz (Ed.). INTECH Open Access Publisher
- [7] Sara Gómez Arnaiz; TFG - Diseño e implementación de un sistema de descomposición automática de señales EMG.
- [8] Quiroga RQ, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 2004; 16: 1661–87.